

**Purdue University**  
**Purdue e-Pubs**

---

College of Technology Masters Theses

College of Technology Theses and Projects

---

4-18-2012

# Virtualization in High-Performance Computing: An Analysis of Physical and Virtual Node Performance

Glendon M. Jungels

*Purdue University*, [gjungels@purdue.edu](mailto:gjungels@purdue.edu)

Follow this and additional works at: <http://docs.lib.purdue.edu/techmasters>



Part of the [Digital Communications and Networking Commons](#), and the [Other Computer Engineering Commons](#)

---

Jungels, Glendon M., "Virtualization in High-Performance Computing: An Analysis of Physical and Virtual Node Performance" (2012). *College of Technology Masters Theses*. Paper 64.  
<http://docs.lib.purdue.edu/techmasters/64>

This document has been made available through Purdue e-Pubs, a service of the Purdue University Libraries. Please contact [epubs@purdue.edu](mailto:epubs@purdue.edu) for additional information.



n o l o g y

## Performance and Virtual Node ce

uirements for the  
in Technology

Report

Date

_____	_____
_____	_____
_____	_____
_____	_____

## Contents

<b>College of Technology</b>	1
<i>Executive Summary</i>	4
<i>Introduction</i>	5
<i>Statement of the Problem</i>	6
<i>Significance of the Problem</i>	7
<i>Statement of Purpose</i>	7
<i>Project Background</i>	9
Relevant History	9
Definitions	10
Review of Literature	11
<i>Project Assumptions</i>	15
<i>Project Delimitations</i>	15
<i>Project Limitations</i>	17
<i>Procedures Employed</i>	19
File-system benchmarking	19
Data from file-system benchmarking	19
Large File Copy	19
Data from file-system benchmarking	24
Small File Copy	24
Data file-system benchmarking	28
Small File Writes	28
<i>Procedures Employed</i>	32
HPC benchmarking	32
Data from HPC benchmarking	33
NPB - BT	33
NPB - CG	34
NPB - EP	35
NPB - FT	37
NPB - IS	38
NPB - LU	39
NPB - MG	41
NPB - SP	42

XHPL .....	44
<i>Conclusions, recommendations and financial implications</i> .....	45
<i>Opportunities for further research</i> .....	46
<b>References</b> .....	<b>Error! Bookmark not defined.</b> 47
<b>Appendices</b> .....	50

## ***Executive Summary***

Advances in computing technology have changed the high-performance computing landscape. Powerful hardware, such as multiprocessor servers, and high-speed, low-latency networking technologies are available at increasingly competitive price to performance ratios. These components, combined with a modern operating system can be utilized to assemble a system capable of such actions as simulating a nuclear explosion, predicting global weather patterns, and rendering a feature length animated film.

Virtualization is a computing process that allows multiple operating systems, or multiple instances of a single operating system, to reside and function on a single computer server. Operating hardware in this fashion offers advantages such as greater flexibility and higher utilization of the server's resources. As a result, it offers possible environmental advantages such as lower power consumption and fewer physical servers to be managed in an organization's data-center.

The following document demonstrates the creation of a high-performance computer using modern hardware and using virtualization technology to efficiently utilize a server's computing capabilities while providing near equivalent performance as compared to equivalent physical server hardware. Resources, such as networked file-systems, and industry respected benchmarking tools are used to accumulate data from performance testing. An analysis of this testing is presented.

The results of this testing show that although creating a high-performance cluster using virtualization is possible, and offers advantages, it does not offer real-world feasibility. Computing performance, as compared to an equivalent physical cluster, is proven to be substantially less in many of the benchmarks utilized, specifically those using high levels of

inter-node communication. Sustained file operations also frequently caused virtual servers to lock up, necessitating a reboot. The real-world implication of these results is that utilizing the resources used in this research, virtual servers should be considered for use in high performance computing where inter-node communication is minimal.

## ***Introduction***

Virtualization has its beginnings from the days when the IBM mainframe was the dominant computing platform. With the introduction of the PC, mainframes and mainframe-type capabilities fell to the technological wayside. Recently, the concept of virtualizing instances of operating systems has taken hold in the PC computing arena. Examples of current virtualization software in use today include EMC's VMWare, SWSOFT's Virtuozzo, the open-source Xen product, Microsoft's Virtual Server and the Linux Kernel based Virtual Machine (KVM). Advancements in server processors and specialized operating system drivers allowing virtualization to take place in hardware on a server have minimized the performance penalty that occurs by virtualizing operating environments.

Previous studies have been completed to show the performance impact of utilizing non-hardware assisted virtualization techniques on high-performance clusters. The impact of such an environment using the open-source Xen software typically constitutes a performance degradation of two to five percent compared to native performance. Hardware assistance for virtualization can be used to compensate. This technology allows a more efficient operating environment in which to deploy what are commonly referred to as high-performance computing (HPC) clusters. This study hypothesizes that HPC node performance, and thus overall HPC cluster performance, can obtain near equivalent performance for CPU bound tasks compared to their native hardware counterparts.

For example, creating a 16 node HPC cluster in which each node represents a virtualized environment bound to one CPU on a 16 processor server will provide performance near equivalent to running 16 physically separate hardware nodes. This virtualized environment provides cluster administrators with potential benefits from fewer physical servers; lower power consumption, lower cooling requirements as well as easier administration for such tasks as deploying new nodes, re-claiming nodes that are not in use and more flexible node resource allocation. In a virtual server environment, resources such as memory and disk space can be assigned without physically touching hardware.

This study builds on these concepts by creating a small (4 node) virtualized cluster on a multi-processor (with hardware assistance) server in which to exercise the efficiencies mentioned above. The results of testing these efficiencies will allow for potential extrapolation of the impact on much larger cluster deployments in an IT environment in which data-center floor space is becoming ever so difficult to find and ever more expensive to build.

### ***Statement of the Problem***

According to Forrester's April 2006 IT Forum, in production environments the number one reason to use this technology is for flexibility. This flexibility is supported by hardware assisted virtualization, such as AMD-V (AMD 2011) and Intel VT (Intel 2011), which allows hardware acceleration of some software operations. One use for virtualization technology is to examine software impacts on clusters without deploying hardware equivalent to that found in an organization's production cluster. (Spigarolo and Davoli 2004)

Even with such hardware acceleration, virtualization does create operating overhead, particularly for I/O operations (Kiyancilar 2006). This leads to an unknown that this research seeks to address. To what degree is performance impacted when using hardware assisted

virtualization to create a virtual high-performance computing cluster?

### ***Significance of the Problem***

Running high-performance computing cluster nodes in a virtualized environment has several implications. One implication is that data-centers do not have to continually expand the physical space required as the data-center grows over time. There are other implications as well. Fewer, but more power efficient hardware means less power consumption and in turn less impact on the environment. This also means that fewer cooling units are needed to cool the data-center floor, allowing for more available space and requiring less consumption. Deploying less physical hardware also demands fewer technicians to keep a data-center operational each day. Collectively, these financial incentives provide compelling justification for the use of virtualization in high performance computing environments.

### ***Statement of Purpose***

A quick browse through any IT trade magazine will provide evidence that the world of virtualization is growing. This capability was created in the 1960's and previously built into high-end computing environments such as mainframes. Today, it is now being built into offerings that are available in the average consumer desktop PC (Huang, Liu, Abali, Panda 2006 ). Mainframe virtualization continues to grow in use and capability as well (Babcock 2007). As data-centers across the US continue to grow, virtualization takes on several important roles.

A recent study published in Information Week indicates that the majority of virtualization is taking place in order to consolidate server workloads onto fewer, but more powerful and resource abundant servers using virtualized servers. This creates several questions to consider when deploying cluster nodes as virtual servers in hardware assisted virtualized environments. Can cluster nodes running as virtualized guests perform CPU intensive tasks as well as their



stand-alone counterpart? Previous studies indicate that using specialized hardware for Input/Output (I/O) operations can minimize performance impacts, but if the answer is no then our question must get more specific (Yu and Vetter 2008). Are there CPU performance penalties in using cluster nodes in hardware assisted virtualized environments? Does the same hold true for memory access and I/O operations such as writing to disks? What are the other advantages and limitations in deploying clusters in such a manner?

The hypothesis to be tested is that utilizing hardware assisted virtualization; high performance computing nodes can perform as nearly well as individually deployed hardware nodes when performing CPU intensive tasks. Hardware assisted virtualization allow for the Virtual Machine Monitor (VMM), the software that provides oversight to the virtualized environment, to pass on instructions previously emulated by software directly to a computer's CPU(s). (IEEE Computing Society 2005). This hypothesis becomes increasingly significant as corporate data-centers run out of physical capacity (floor space, electrical power capacity, cooling capacity). Running multiple cluster nodes on a single, multi-processor/multi-core server can aggregate equal amounts of computing power into a smaller amount of data-center space than other methods such as utilizing blade-server technology.

In order to test this hypothesis, a hardware-assisted, virtualized four node cluster will be built for the purpose of comparing computational efficiency of virtualized clusters to that of native cluster environments. The testing environment will be using the Linux operating system as it is the predominant high-performance operating system in use by clusters today (Top500.org 2011).

In short, the purpose of this study is to determine the viability of using virtualized cluster computing nodes in place of traditional nodes. In addition, the study will serve to determine if

the *potential* exists for cost and management overhead savings while retaining an equivalent level of computing performance when deploying cluster nodes as hardware-assisted virtualized servers.

## ***Project Background***

### **Relevant History**

Completing searches in research databases yields few, but applicable, results at the time this initial research was completed. In 2004, instructors at the University of Bologna, Italy, constructed the Berserkr Beowulf cluster. This cluster utilizes a software virtualization (User Mode Linux) approach which allows for multiple Linux kernels to run in user-space (the area of memory in which most applications are run). The primary purpose of Berserkr is not that of performance, but rather for testing, teaching, security (of resource assignment) in a low cost environment. Specifically, this virtualized cluster is used to teach parallel programming methods in a computer science curriculum without the associated high cost of a traditional computing cluster (Spigarolo and Davoli 2004).

Faculty from Ohio State's Computer Science and Engineering department teamed up with researchers from the IBM T. J. Watson Research Center to propose that high-performance computing clusters deployed in virtualized environments have advantages over other deployments. Among the proposed advantages is ease of management, customized operating system and advanced system security by enabling only the services necessary for a program to run (a point more applicable to environments in which computing resources are shared between departments, etc). This study includes the use of the Xen virtualization technology. It acknowledges several limitations with the architecture, specifically with input/output operations. By utilizing custom software, the impacts of these limitations were minimized and enabled their

project to use a high-speed/low latency interconnects called Infiniband, rather than traditional Ethernet for communications.

Although not specifically mentioned, the timing of this study, as well as the types of hardware (no model numbers), leads one to believe that the authors did not utilize hardware assisted virtualization in creating the cluster in their study. Useful cluster performance benchmarking tools, such as NAS Parallel Benchmarks, were mentioned that will be used in this study (Huang, Liu, Abali, and Panda 2006).

## Definitions

**Virtualization:** an abstraction of computing resources by hiding the physical computing resources and making it appear as a logical unit.

**Virtual Machine Monitor (VMM):** also known as a hypervisor. This is the platform that allows for multiple concurrent operating systems to run on the same physical hardware.

**Hardware Assisted Virtualization:** this is abstraction of computing resources performed at the hardware (CPU) level. This type of abstraction offers increased performance as hardware intercepts, and performs, hypervisor system calls rather than being emulated by software.

**Cluster:** in the context of this research paper, a cluster is a group of machines that work together to perform analysis of data in parallel, thus increasing the speed at which the analysis takes place.

**Parallel Processing:** the process of breaking data into pieces and spreading the analysis over a number of machines to decrease the amount of time to complete the analysis.

## Review of Literature

This study represents a combination of studies in clustering, virtualization, and performance relative to running a cluster on standard hardware (not in a virtual environment) and its associated business implications. There is the engineering, or very technology specific, application of creating a cluster in a virtual environment. The research for literature needed to be representative of this view. To fulfill this research need, the three search databases that were utilized were: the Purdue University ACM Portal, Compendex and IEEE Xplorer. There is also a business, or real world implementation and benefit. Trade organizations such as Forrester and Gartner provided the necessary business outlook on the emerging technology known as virtualization, and to a lesser extent on clustering. Terms used to search these databases included: Xen, clustering, virtualization, benchmarking tools, Linux, technology, techniques, computing, hardware assisted virtualization, and various combinations of these terms. As some searches in these databases yielded few results, a search utilizing more open search tools, such as Google, was necessary to provide sufficient avenues for further research in this area.

Similar searches in the distinctly different style databases provided unique perspectives on the technologies of this study. These searches also provided much different summary areas than the search terms would have indicated. These can be categorized as follows: virtualization technologies and general impact on information technology, virtualization and utilization with clustering technology (specific to this study), and performance/benchmarking software representing a potential tool-set upon which to quantify how closely a virtualized cluster compares to clusters utilizing traditional deployments of rack mounted and blade servers.

History suggests that the success or failure of a technology is often dictated by the entities backing it, rather than the merits of simply the technology on its own. With proper

support, a technology can gather the moment to garner the attention of investment and finance entities. According to studies, virtualization is an example of this type of technology.

Forrester conducted research in late 2005 to determine adoption trends of virtualization. This research included 56 North American companies with 500 or more employees. In addition, the firm conducted a roundtable discussion at its April 2006 IT Forum. The results support virtualization technology as one that will have an ongoing impact on IT and business operations. Sixty percent of the respondents in the combined roundtable and survey reported use of some type of virtualization technology (Gillett and Schreck 2006). In many instances, virtualization technology is used for testing and development. Due to the ease of setting up a virtualized cluster to mimic a production cluster, but on less capable hardware, this strategy is also used to examine potential software impacts on high-performance computing clusters (Spigarolo and Davoli 2004).

The Forrester research also indicated that virtualization is being deployed in production environments as well for a variety of purposes such as file and prints sharing, web serving, serving custom applications, infrastructure roles such as DNS and DHCP, and a multitude of other purposes. The primary reason cited for using virtualization is flexibility, followed by consolidation and disaster recovery purposes. Other tangible and measurable benefits mentioned included floor space savings in the data-center, reduced energy consumption and reduced cooling needs.

The Forrester results are supported by The Wall Street Journal on March 6, 2007 entitled “Virtualization is Pumping Up Servers—Software that Enables Use of Fewer Machines May Cut Hardware Sales.” As a demonstration, the article describes a company that has consolidated servers using virtualization technology and eliminated 134 servers, with more than three dozen to

be phased out by the end of 2007. Using technology similar to that utilized in HPC clusters, shared computing capacity was spread across the server farm using Virtual Iron from Virtual Iron Software, Inc. (Lawton and Clark 2007).

The impacts virtualization has on saving data-center floor space, energy consumption and reduced cooling are quite real. Traditional blade servers and other high density server deployments require massive amounts of power per square inch of data-center floor space. This equates to a fewer number of servers per rack in a data-center in order to maintain required electrical and cooling needs.

A study conducted by International Data Group (IDG) and Hewlett Packard concluded exactly this. Using AMD's newest sixteen core Opteron server processors, the group was able to virtualize workloads, maintain equivalent performance levels, and cut power consumption. The white paper cites being able achieve the following (IDG 2011):

- Up to 50% greater throughput in the same power and thermal footprint
- Load 33% more virtual machines per server
- Fit more servers within the existing power allotment

As industry and deployment dictate, it appears that virtualization is a legitimate technology that is being increasingly deployed in production environments today. This leads one to question if this same virtualization technology has been used in a HPC cluster.

Other sources of information specific to virtual clusters include a study at the National Center for Supercomputing Applications that focuses on creating on-demand clusters, as well as a study from Argonne National Laboratory suggesting the use of virtual clusters in support of national grids in an on-demand fashion as resources allow such allocation. Neither of these two additional studies utilized hardware assisted virtualization as the technology had not yet been

released at the time the studies were completed.

This study relies heavily on comparing performance of cluster nodes deployed as virtual servers to cluster nodes deployed as hardware servers. It is appropriate that a review of benchmarking tools and literature be completed. Unfortunately, searches in the preferred databases, such as IEEE Xplorer, did not provide suitable results and thus justified the use of a more open search engine using Google. The results returned were overwhelming in number. The research portion of the study consumed the greatest amount of time. This research did uncover a number of tools to be used in completing this study. These tools include both commercial and open-source (freely available) benchmarking tools.

The Ohio State/IBM virtualized cluster study made use of Numerical Aerodynamic Simulation (NAS) Parallel Benchmark. This package was developed by NASA at the Ames research center to test the efficiency of parallel processing systems, specifically those that are used for Computational Fluid Dynamics. NASA developed this package of benchmarks to be as generic as possible in order to provide a generic set of tools applicable across a variety of architectures (Bailey, Barszcz, Barton, Browning, Carter, Dagum, Fatoohi, Finebeg, Frederickson, Lasinski, Schreiber, Simon, Venkatakrishnan, Weeratunga 1994). NAS Parallel Benchmark tests CPU, memory, I/O, and network response and outputs the results into a comma-delimited text file that can be easily imported into an Excel spreadsheet for more detailed analysis and graphical presentation.

Other tools identified on sourceforge.net include: LMBench, Procbench, Sysbench as well as commercial products such as Sisoft's SANDRA product and the SPEC products from Spec.org. The focus of this research will be on utilizing freely available, open technologies.

This literature review concludes that previous work has been completed in the area of

utilizing virtualization techniques in high-performance clusters. Because of the relative newness (released roughly in the three months) of the technology, a virtualized cluster using hardware assisted virtualization could not be located. This study will build on the excellent work from the Ohio State/IBM study and utilize a number of benchmarking tools found while completing the review, including the respected NAS Parallel Benchmark.

### ***Project Assumptions***

This study defines clustering as it applies to high-performance computing. Specifically, it will address clusters as deployed in a parallel computing environment. It will not be addressing alternate definitions of clustering, as it applies to high-availability and automatic fail-over of computing resources.

Hardware, software and underlying techniques to make them work together improve with time. Virtualization software, techniques and hardware are not exceptions to this paradigm. Problems encountered in completing this project would likely not be encountered if completed on updated hardware and using modern virtualization software/techniques.

### ***Project Delimitations***

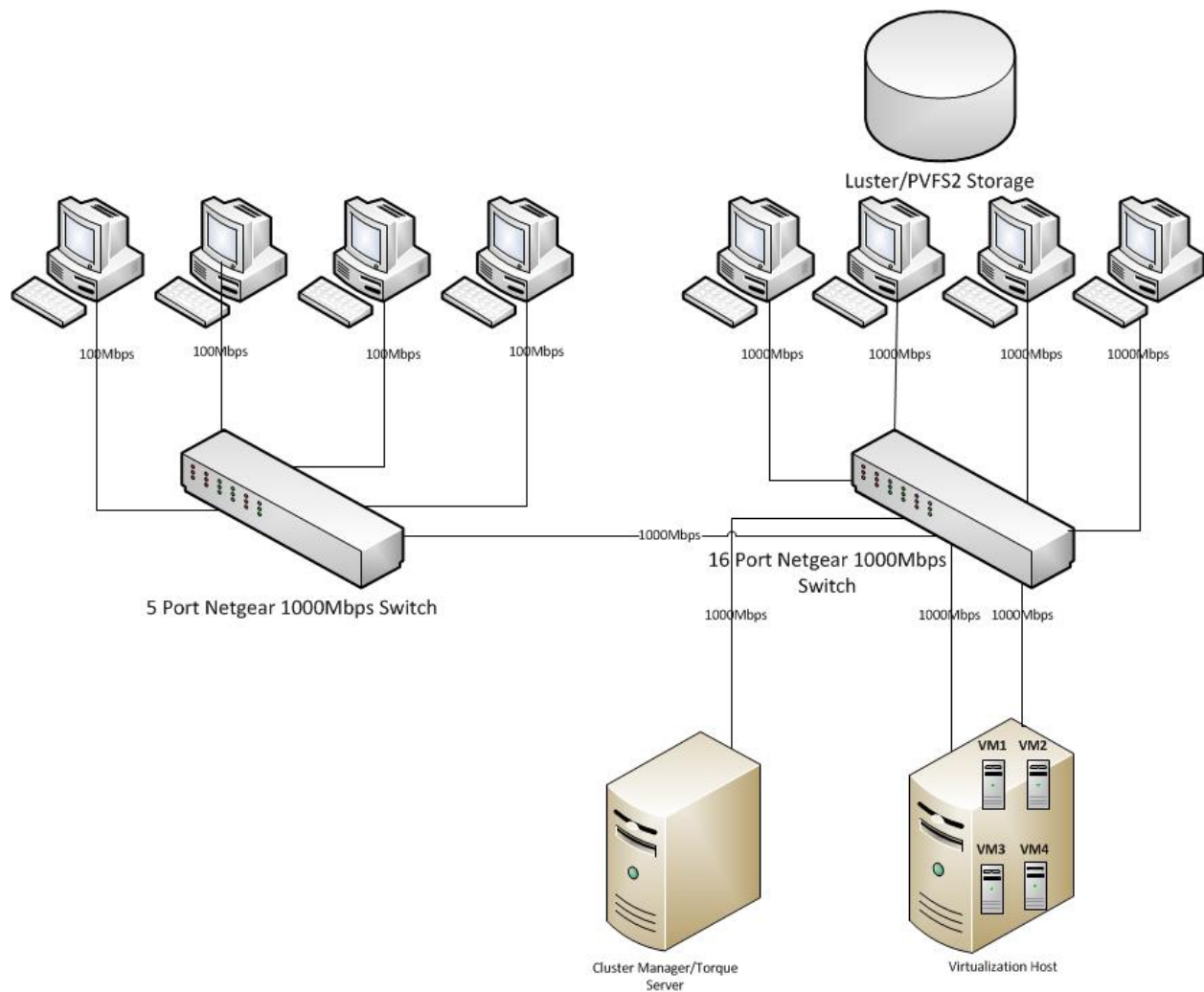
The following software resources were used to complete this directed project:

Open Source Cluster Application Resources (OSCAR)	5.1.0	Cluster imaging and resource tool
Red Hat Enterprise Linux	5.0	Cluster node operating system
Citrix XenServer	6.0	Virtualization host software
Torque	2.1.8	Cluster resource manager
Maui	3.2.6p19	Cluster workload manager



Whamcloud Luster	1.8.7	Parallel file-system
Parallel Virtual FileSystem (PVFS)	2.0	Parallel file-system
Network File System (NFS)	3.0	Distributed file-system
Red Hat Linux Kernel	2.6.18-274	Operating system kernel patched to support Lustre

The following network diagram shows the network deployed to complete this project:



Based on a review of the literature, this project differs from others that are similar. Similar projects conducted utilized a single virtual processing node on each of several servers. This project utilized all virtualized processing nodes (each assigned to one processor) on a single server. This leads to differences such as resource sharing contention for access to file-systems that host the virtual machines as well as access to shared hardware such as the network controller.

This project utilized full virtualization for the virtual node rather than an alternate technique called para-virtualization that requires a modified guest kernel in order to run on the virtualization host. The earlier para-virtualization technique does not take advantage of hardware assisted virtualization thus not suitable for this study. It also requires significant kernel patching to utilize and one goal of this project was to use a kernel as close to stock as possible.

The Linux operating system used in this study utilizes a predominantly standard Linux kernel and makes use of the hardware's virtualization processing technology to operate unaltered. A single difference in the kernel was the application of a patch for the parallel file-system testing portion of this project. This was necessary to implement the Luster file-system on both the file-system side, as well as the client side. This study did not address utilizing multi-CPU virtual machine configurations.

### ***Project Limitations***

The virtualization host machine used in this study contains two processors and four processing cores. This leaves one machine at a given time sharing CPU time with the host. The physical machines (IBM desktop computers) used in the study are able to address 512 Megabytes of RAM and run at a processing speed of 1.8 GHz.

The virtualization host utilized two AMD Opteron 2212 processors, each with two cores.

Each core on the virtualization host utilizes a processing speed of 2.0 GHz. Total virtualization-host RAM of 4096 Megabytes limits each virtual machine to a maximum of less than 1024 Megabytes. Some RAM is needed by the host for processing. The version of Xen used in this project, XenServer 3.2, does not include the ability to tie a virtual host to a specific CPU. This version of XenServer also limits the number of virtual machines to 1 virtual machine per CPU, or core for multi-core machines, for a maximum of four running virtual cluster nodes.

The testing for this project consists of two associated lines of tests. The first line of testing completed benchmarks the performance of completing file-system operations on two parallel file-systems and one networked file-system by virtual cluster nodes and physical cluster nodes in independent tests. The second line of testing was completed using two well-known benchmarking programs, NAS Parallel Benchmarks and XHPL. The procedures, data, and results for the file-system testing are presented first, followed by the same format for the two benchmarking utilities.

## Procedures Employed

### File-system benchmarking

The file-system testing consists of three separate tests. The first test copies a large file from the remote file-system to a location on the local node. The second copies a large number of files from a local node location to the remote file-system. The final test writes new files to the remote file-system. These tests were completed using PERL scripts, available in the appendix, on each node specific to each test, differing on each node only by the target remote file-system and folder names. Each test runs a series of 35 iterations. The time necessary to complete the iteration was logged to a text file for later analysis. The tests were completed with one, two, three and four simultaneous nodes performing the test. Only the results for four nodes are shown.

### Data from file-system benchmarking

#### Large File Copy

Figure 1 – Lustre Large File Copy

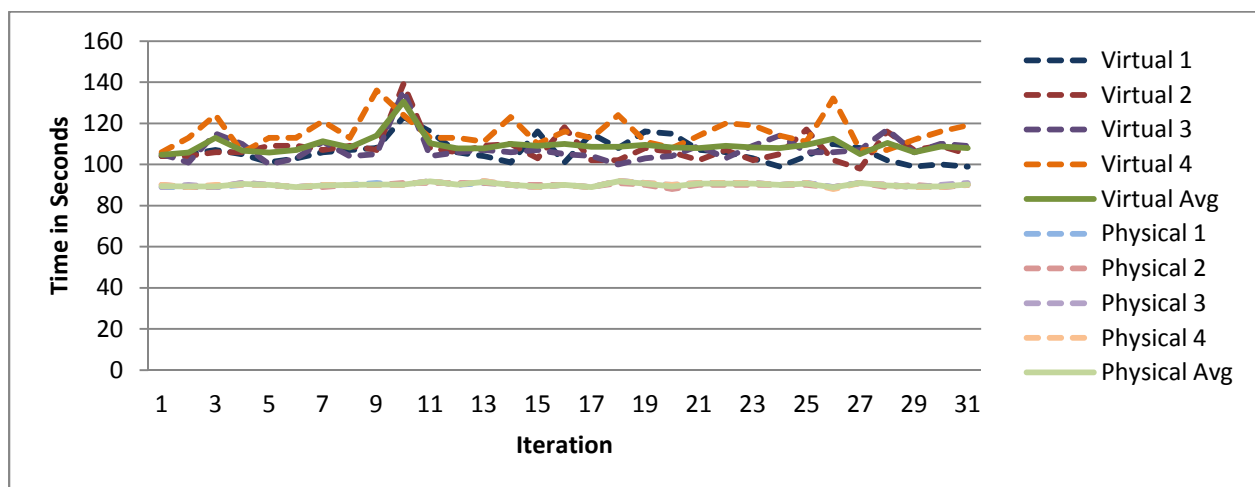


Figure 2 – PVFS Large File Copy

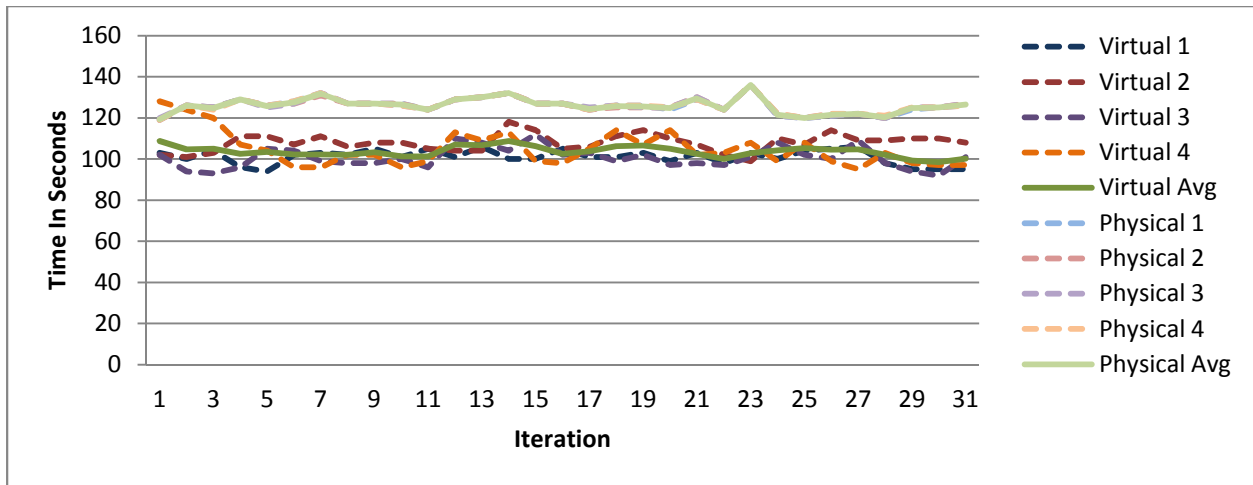


Figure 3 – NFS Large File Copy

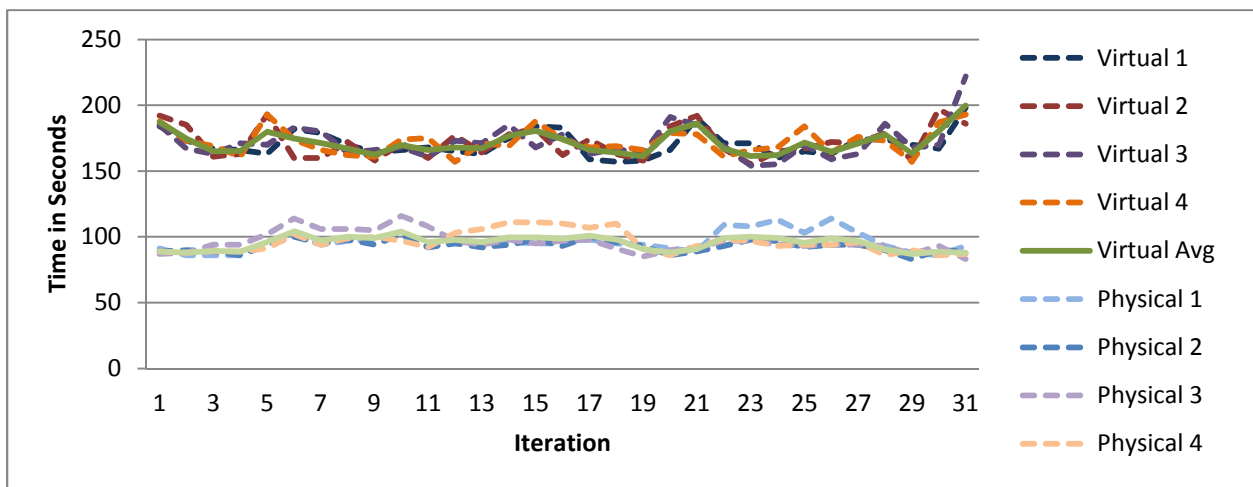
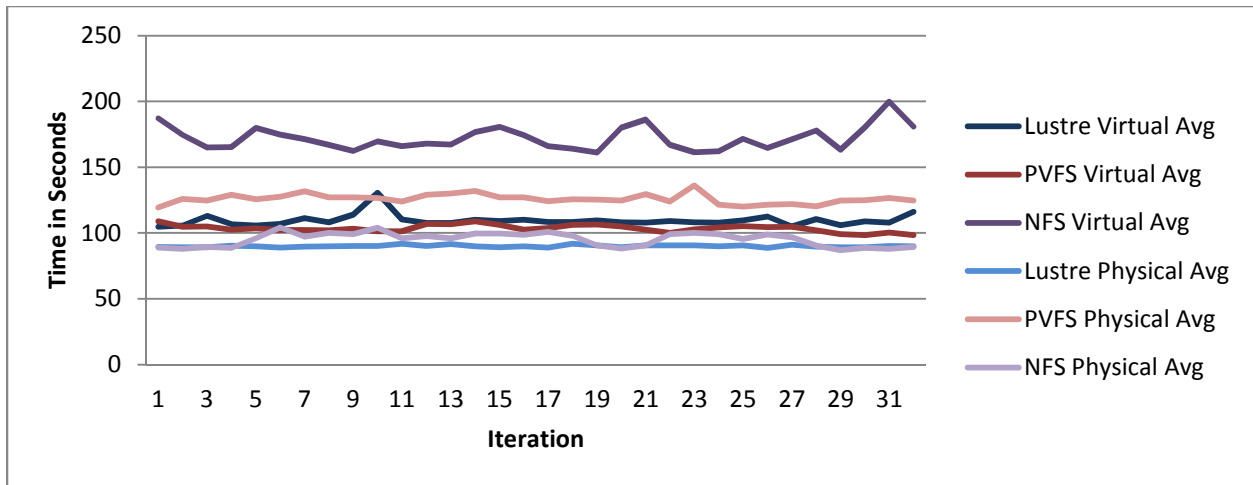


Figure 4 – Averages Large File Copy



### Results – Large File Copy

This test was conducted for multiple purposes. Its first purpose is to compare the amount of time it takes to copy a large file of known size to a local file-system on a cluster node. Second, it compares two commonly deployed parallel file-systems used in high-performance computing clusters. Third, it compares the copy time of physical machines to that of the virtual machines. The final purpose is to compare the parallel file-systems to a commonly used conventional network file-system. The two parallel file-systems used in this test were Luster and Parallel Virtual File System version 2 (PVFS) and the single network file-system used was Network File System (NFS).

The script for this test (Appendix *Figure 22*), copies the test file from the target file-system (Lustre, PVFS, NFS) to a location on the client local file-system over a series of 35 iterations, timing and recording the copy time to a file for later analysis. This was done with a single client, two clients, three clients, and four clients to record the impacts of increasing load. The same procedure was conducted on physical cluster nodes Figures 1-3 show each file-system with four concurrent copies. Figure 1-4 displays a summary of the averages for each file-system

as accessed by both physical machines and virtual machines.

Luster was the first file-system to be tested. Figure 1 shows an immediate difference between the copy times for the virtual machines and the physical machines. The physical machine copy times show little variance. The virtual machine copy times are both longer and more erratic. There are visible peaks and valleys visible over the course of the 35 iterations that are fewer and to a lesser degree than the physical machines. There are several possible explanations for the peaks and valleys. Each physical machine has its own 100Mbps network card. This is not the case with the virtual machines as each virtual machine shares a 1000Mbps network card with another virtual machine. The 1000Mbps connection offered by the virtualization host offers little advantage to the virtual machines as each is constrained by a 100Mbps driver. Another potential explanation for the higher copy times for the virtual machines is the simultaneous writes to the shared local storage housing the virtual machines. Other possibilities include contention for host processor cache, contention for interrupt requests and contention for disk buffers on the host local storage.

The script (Appendix *Figure 23*) copies the same file to the PVFS file-system. The results for PVFS are presented in Figure 2. Like the copies from Lustre, the copy times from PVFS vary between the physical machines and the virtual machines. The physical machines show more variation than with Lustre while the virtual machines copy times are less erratic. Unlike the copies on Lustre, the file copies on PVFS complete faster on the virtual machines than on the physical machines. With PVFS, peaks and valleys occur much more simultaneously across the machines, both physical and virtual. A closer investigation is necessary to determine the reasons PVFS and Lustre display inverse tendencies with copying large files on virtual machines versus physical machines.

The script (Appendix *Figure 24*) copies the file to the NFS file-system. Figure 3 depicts the final file-system used in this test, NFS. Although not a parallel file-system, NFS performs very well for this test. The variance pattern between the virtual machines is comparable to that of the physical machines, though the copy times are greater. The overall copy times also correspond closely to that of the parallel file-systems. For HPC workloads requiring the movement of large files from a shared file-system to local storage, NFS is as capable as both Lustre and PVFS. Years of development and use in a large variety of workloads have contributed to a stable and well performing storage file-system.

Figure 4 provides an overview of the three file-systems. The file copies completed on the physical machines did so faster than any file-system on the virtual machines. Lustre shows the best performance overall and is the best performing file-system for the virtual machines. NFS on the physical machines provides very similar copy times to Lustre, but shows the highest copy times and also the greatest variance in copy times for the virtual machines. The performance of PVFS on virtual machines is the highest while PVFS on physical machines is close to mid-way between Lustre on the physical machines and NFS on the virtual machines. These are the results for a small number of machines. As expected with each file-system, as load increased so did the times to complete an iteration of the test. Lustre, PVFS, and NFS all performed similarly on the physical machines. These results indicate that under this type of workload, parallel file-systems offer little advantage, though with loads surpassing those created by this test, the advantage would become apparent.



## Data from file-system benchmarking

### Small File Copy

Figure 5 - Lustre Small File Copy

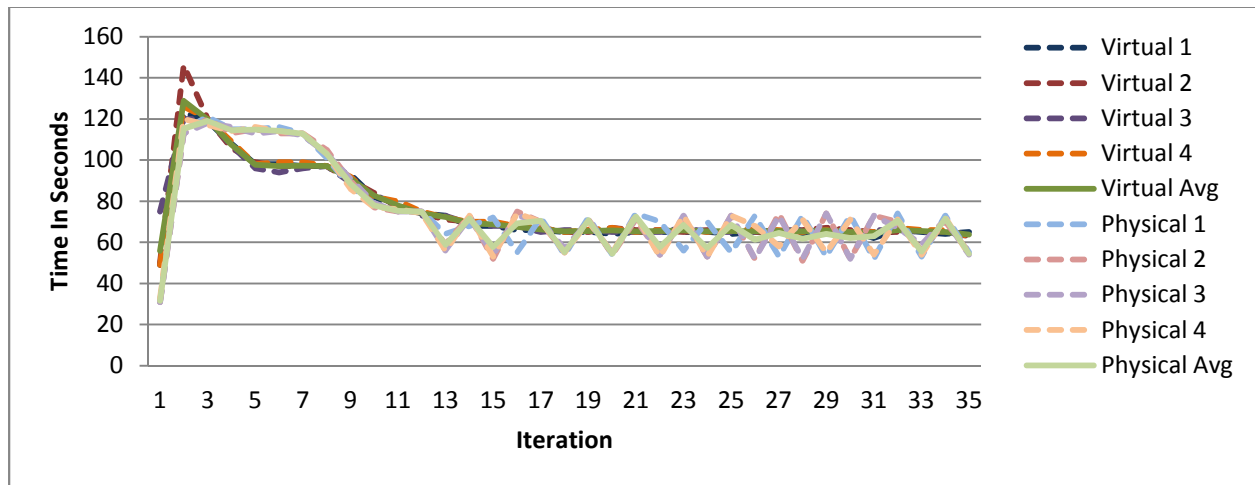


Figure 6 - PVFS Small File Copy

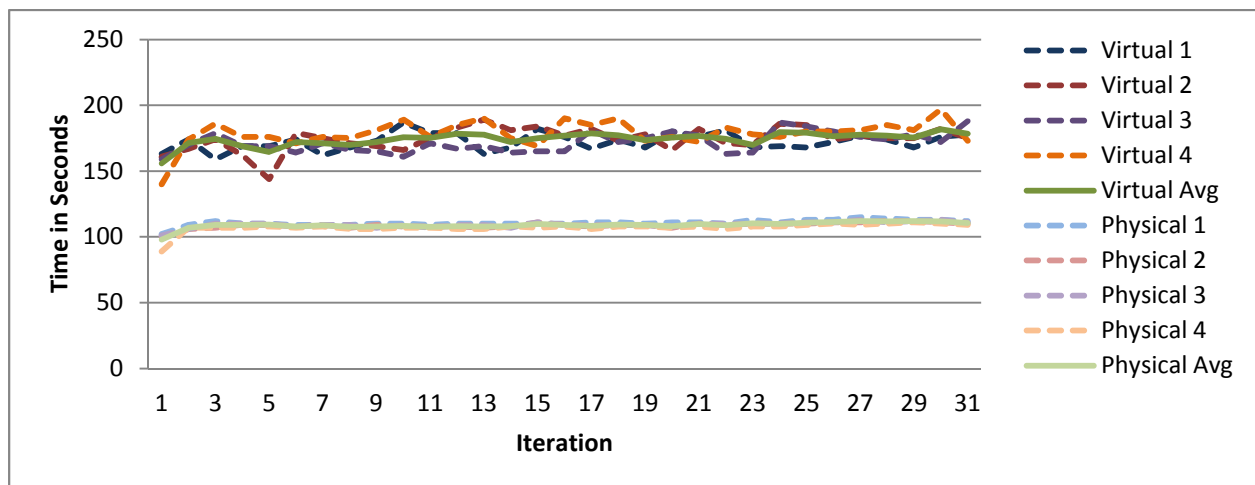


Figure 7 - NFS Small File Copy

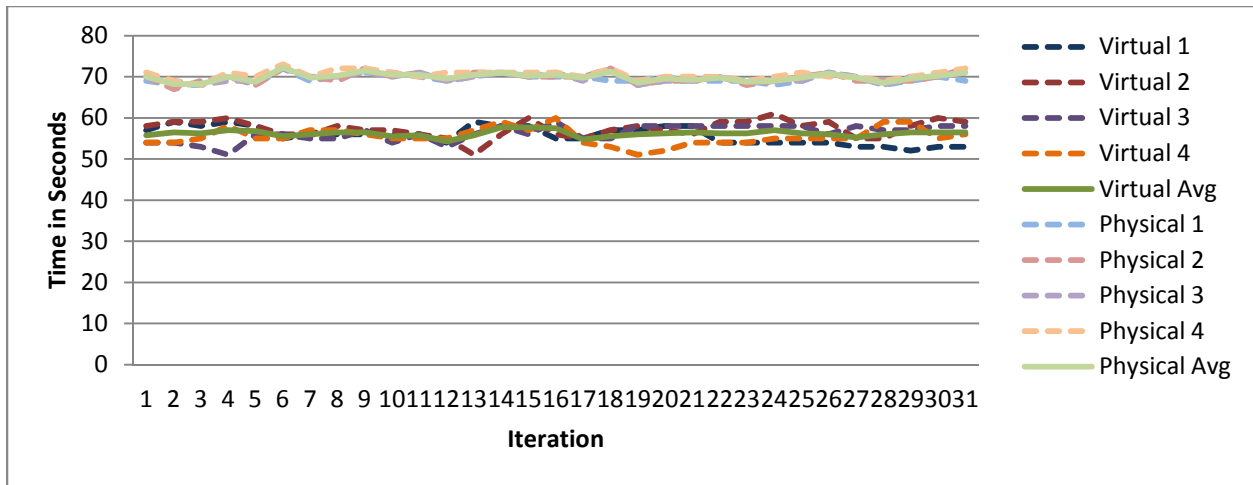
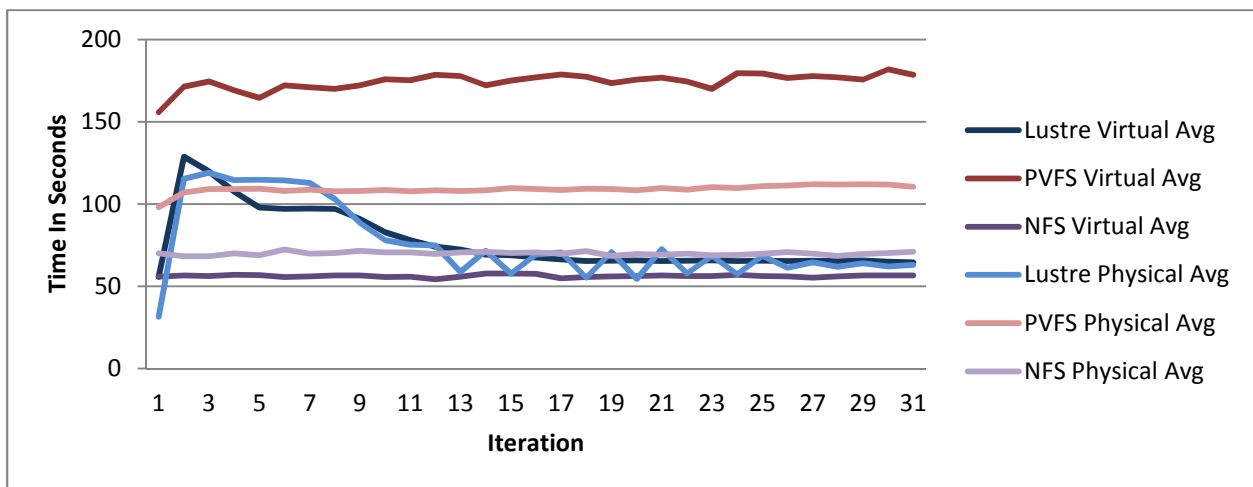


Figure 8 - Averages Small File Copy



## Results – Small File Copy

The small file copy test copies 1000 one kilobyte files to a folder on the target file-system for 35 iterations. Each machine copies to a separate folder and the time it takes to complete the iteration is written to a file on the local machine file-system. This test was conducted to compare the copy times of virtual machines to that of the physical machines for target file-systems on Lustre, PVFS and NFS. Like the large file copy described earlier, this test was completed with one client, two clients, three clients and four clients running simultaneously. The results of the

tests are presented in Figures 5 – 8.

The script in Appendix *Figure 25* performs the small file copy test for Lustre. Figure 5 show the results of the small file copy test on the Lustre file-system. The graph depicts some unusual results at the beginning that taper off approximately one-third of the way through the thirty five iterations of the test. Unlike the previous test, copying files to the target places continuous load on the meta-data server for the parallel file-system as new files are added. Near iteration 11, the copies stabilize and this continues until the final iteration. This is observed by tests on both the physical machines and the virtual machines. Unlike previous tests, the physical machines show variability in copy times resulting in the saw-blade look of the graph. The results also show that the copy times are almost identical within the respective groups. At various points, most notably for the virtual machines, the lines appear very close to being a single line, even with the jagged pattern of the physical machines.

Appendix *Figure 26* is the script used to perform the small file copy test to PVFS. The results of the small file copy test for PVFS are shown in Figure 6. With PVFS, the graphs again look closer to that of the large file copy. Also once again, the virtual machines show increased variability that is not present with the physical machines. With PVFS, it is the physical machines that appear to be a single solid line. Copies on the physical machines also complete faster, unlike the large file copy in which the virtual machines perform better. One key difference between the copies on Lustre and those on PVFS is the time it to complete each iteration. The copies on PVFS take considerably longer for both the physical machines and the virtual machines.

Appendix *Figure 27* is the script used to perform the small file copy test to NFS. Figure 7 shows the results of the small file copy test on NFS. Like PVFS, NFS shows some degree of variability in this test. This is most visible with the virtual machines, though also present with

the physical machines as well. Unlike the previous large file test with NFS, the virtual machines complete these tests faster than the physical machines. A final observation is that the slowest copies by the physical machines on NFS are as fast, or slightly faster, than the fastest copies on Lustre and the fastest virtual machine copy on NFS is nearly half the time of the fastest copy on PVFS. Like the previous large copy test, the advantages of using a parallel file-system are not realized by these tests with a small number of clients.

Figure 8 presents a summary of the three file-systems for small file copies on both virtual machines and physical machines. NFS on the virtual machines proves to be the best combination running this test, followed closely by Lustre on physical machines. Once again, the unusual beginning of the tests for Lustre on both the virtual machines and the physical machines is visible. As Lustre stabilizes, the times become an almost three way tie between Lustre on physical machines, Lustre on virtual machines, and NFS on physical machines. Like previous tests, PVFS does not excel in this test and shows the two overall highest copy times. With limited load, the simplicity of NFS proves again that it is a capable file-system.

## Data file-system benchmarking

### Small File Writes

Figure 9 - Lustre Small File Write

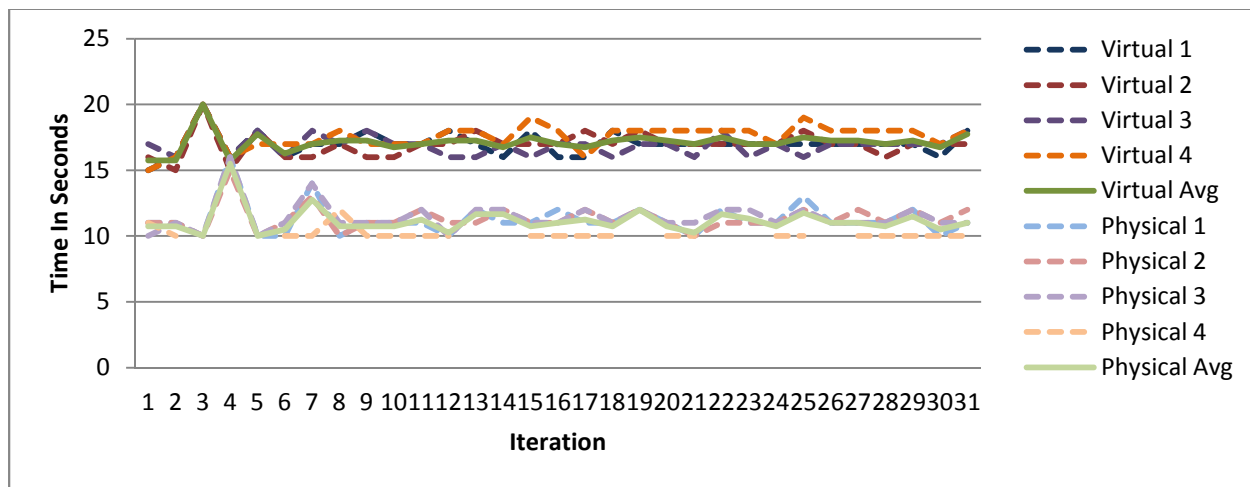


Figure 10 - PVFS Small File Write

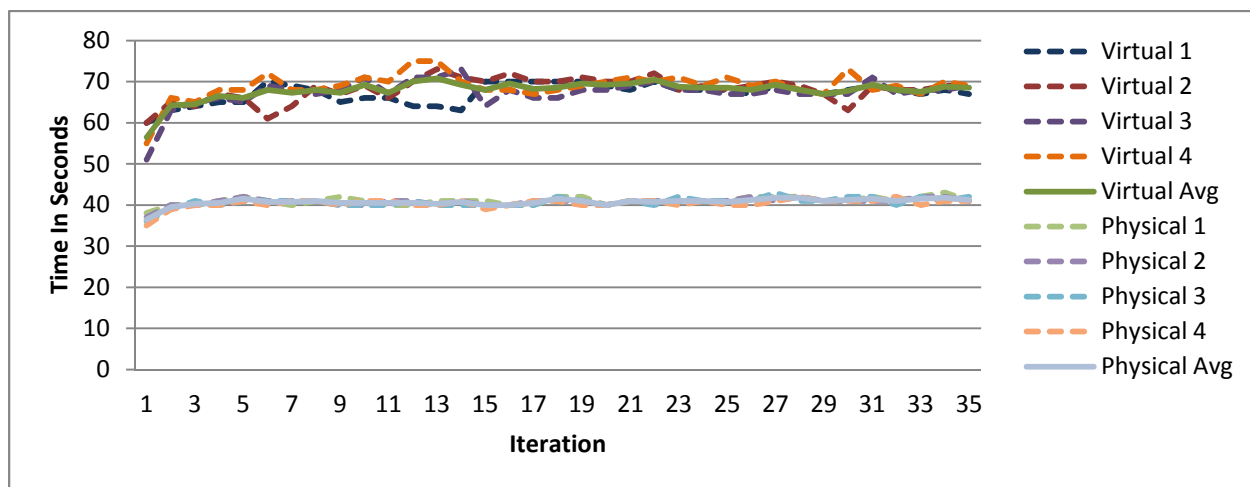


Figure 11 - NFS Small File Write

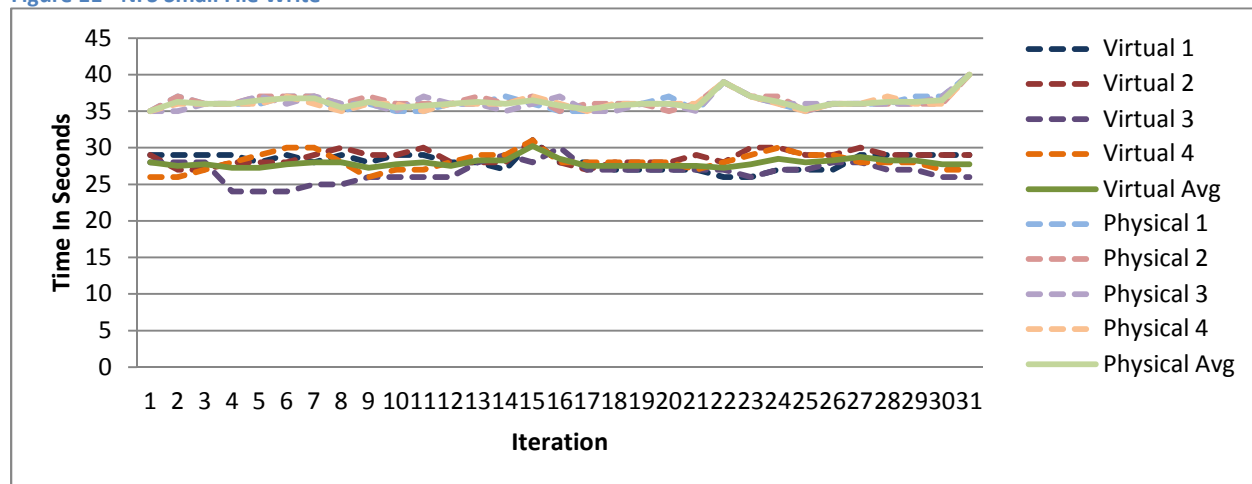
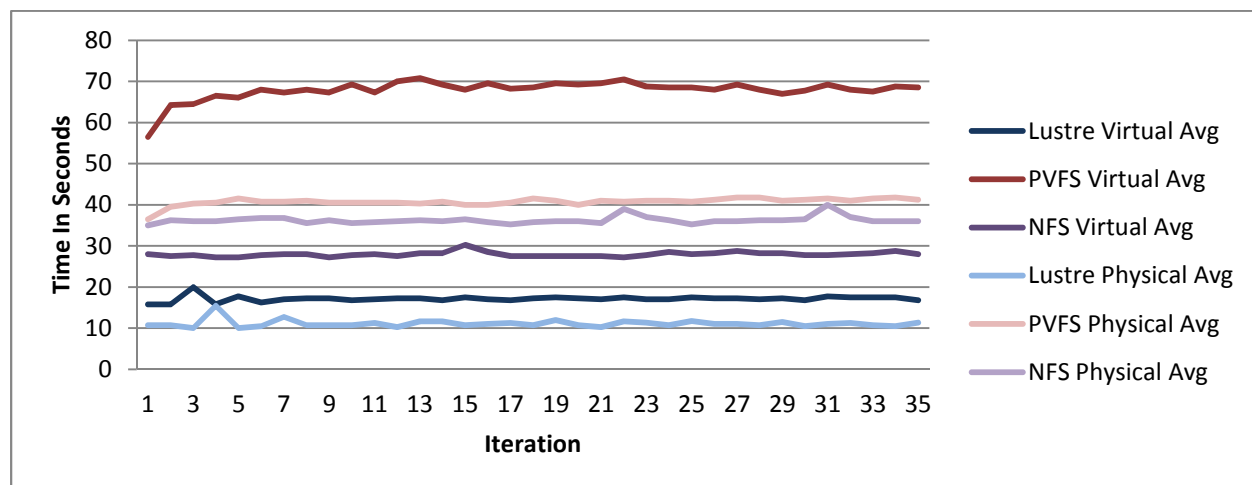


Figure 12 - Averages Small File Write



## Results – Small File Writes

The last set of tests performed consists of writing one kilobyte files 1024 times in a single iteration and repeating this for 35 iterations. This test results in a total of thirty five megabytes written to the target file-system. The iteration was timed and the result logged to a text file on the local file-system. This test was conducted to compare how quickly a physical or virtual machine could write to a remote file-system. Operations such as this are common in high-performance computing calculations where the results are then aggregated into a final format as a

result of the processing. Analyzing the write times serves as an indicator of the speed in which a virtual machine can write a file in comparison to that of a physical machine as well as serve as an indicator of which of the file-systems tested is the most efficient at writing small files.

The script in Appendix *Figure 28* was used to complete the small file write test on Lustre. Figure 9 shows the results of the small file write test on the Lustre file-system. The first several iterations show an unusual pattern, similar to that which was observed in the small file copy test for Lustre previously discussed. After this small number of iterations, the write times even off and remain relatively consistent for the duration of the testing. Both the physical machines as well as the virtual machines display a small bit of variability, though not exaggerated like in other tests. Previous tests have shown that the physical machines complete the tests faster and this is the case with this test as well. There are visible gaps in the graph. During these iterations, present only in the data for the physical machines, the machine was unable to write the file to the Lustre file-system. During those same times, other machines continued to write uninterrupted, but with higher write times than when this event is not occurring. Rather than being unable to write files for periods of time, the virtual machines display a different observable behavior. Flat lines on the graph indicate that those iterations took the exact number of seconds for those iterations. The timer does utilize whole seconds as the unit of measure, but this behavior was not observed in other tests.

Appendix *Figure 29* is the script used to complete the small file write test on PVFS. The results for PVFS for writing small files are shown in Figure 10. The writes for PVFS show a kind of “ramp up” behavior for both the physical machines for iterations one through three. From iteration three forward, the write times are much more consistent with the physical machines again completing the iterations more quickly. The physical machines write to the

PVFS file-system with times very similar between each machine. Like previous tests, the virtual machines show more variable times between iterations, with one a section of approximately ten iterations where the times were very close. There is no missing data for the writes to PVFS as was observed in the file writes to Lustre.

The script in Appendix *Figure 30* was utilized to complete the small file write test on NFS. Figure 11 displays the results for small file write test to the NFS file-system. Like the previous test on PVFS, both the physical machines and virtual machines were able to complete the writes for all iterations. NFS does display write behavior similar to that of Lustre, showing write times of consistently the same time. Unlike Lustre, this is observed for multiple virtual machines, at times concurrently. Using NFS as the target file-system, the virtual machines also recorded lower write times than the physical machines. Write times on NFS were greater than on Lustre, but lower than write times on PVFS. Though more prevalent with the virtual machines, variability in write times is minimal and there primarily in the first eleven iterations.

Figure 12 shows the summary results for small file writes for all three file-systems from both the physical and virtual machines. Lustre shows the lowest write times per iteration for both physical machines and virtual machines, but as indicated previously, also had write failures from a physical machine during testing while others were able to continue. Writes to NFS from the physical machines and virtual machines show the next best write times per iteration, with all iterations completed without a failure to complete a write. PVFS on physical machines and virtual machines display the slowest write times of the file-systems tested. Like NFS, PVFS completed all iterations without a failure to write, though with a higher write time.



## ***Procedures Employed***

### **HPC benchmarking**

As with the file-systems tests, scripting, as shown in Appendix *Figures 32 – 34*, was used to automate running the benchmark. These scripts controlled the timing and submission of the individual benchmarks to the cluster resource manager for assignment and completion. Upon completion of all HPC benchmarks, a PERL script was run against the output files to consolidate the results into a single text file that was later imported into Excel for analysis.

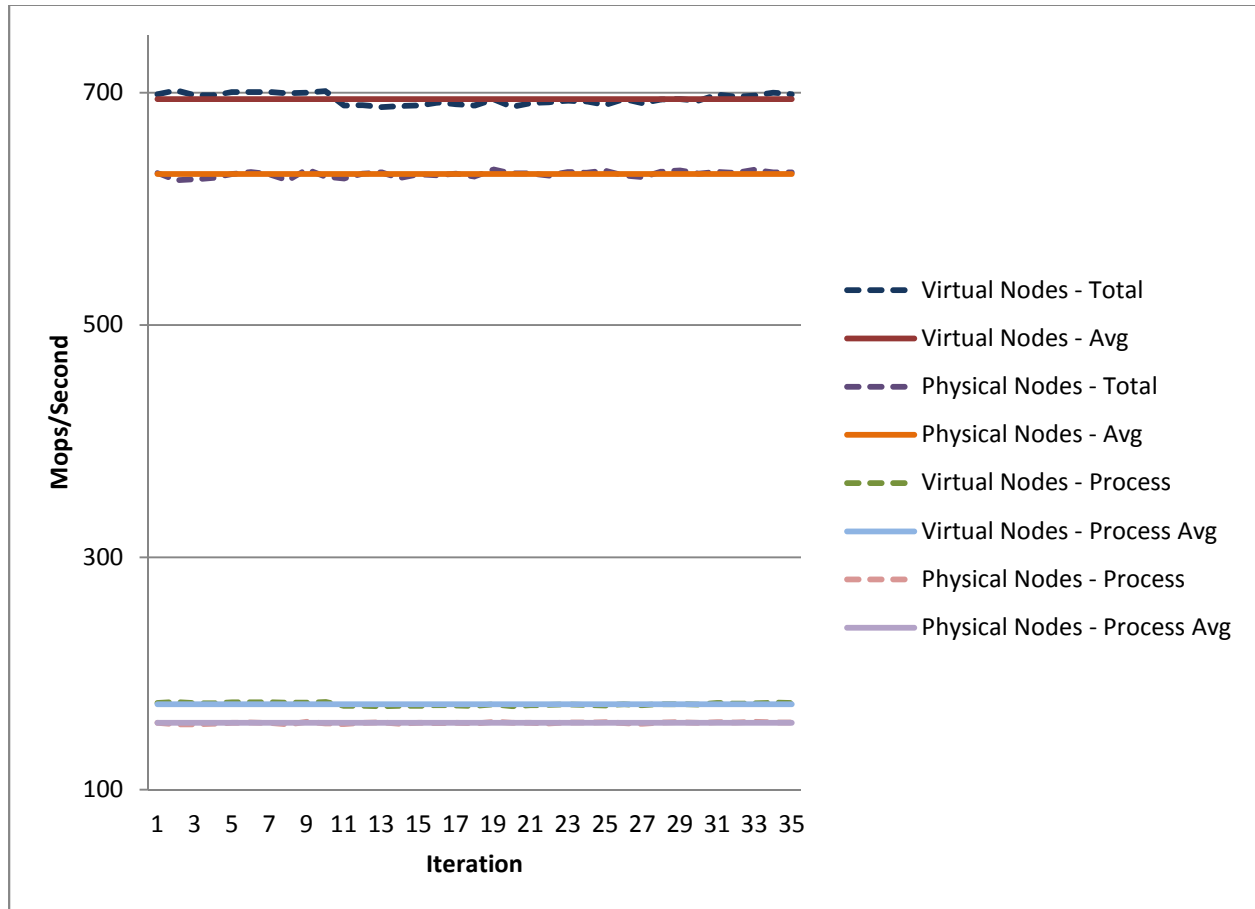
#### ***Benchmark Description – NAS Parallel Benchmarks***

NASA Advanced Supercomputing (NAS) is responsible for the creation of the NAS Parallel Benchmarks. This small set of parallel applications was written at NASA Ames Supercomputing Center as a way to benchmark new high-performance computers being deployed. These applications utilize Computational Fluid Dynamics (CFD) equations commonly in use as part of NASA research. Together, the applications provide a generalization of the performance a new supercomputer can be expected to achieve when applied to real-world problems. The eight benchmarks test a variety of characteristics including memory access, node to node communication and processor performance. Each benchmark has a number of classes that can be utilized. The classes differ in the problem size utilized. For classes A-C, problem sizes increase by four times over the previous class. Classes D, E and F, used for testing very large supercomputers, utilize a step of sixteen times over the previous step. The W class is present, but now deprecated and the S class is intended to provide a quick test of functionality. This research utilized class B for all benchmarks presented. This problem size kept the cluster working longer than the class A benchmark, but did not exceed the memory capability of the hardware utilized as with the class C benchmark

## Data from HPC benchmarking

### NPB - BT

Figure 133 - NPB BT Benchmark



### Results – NPB BT

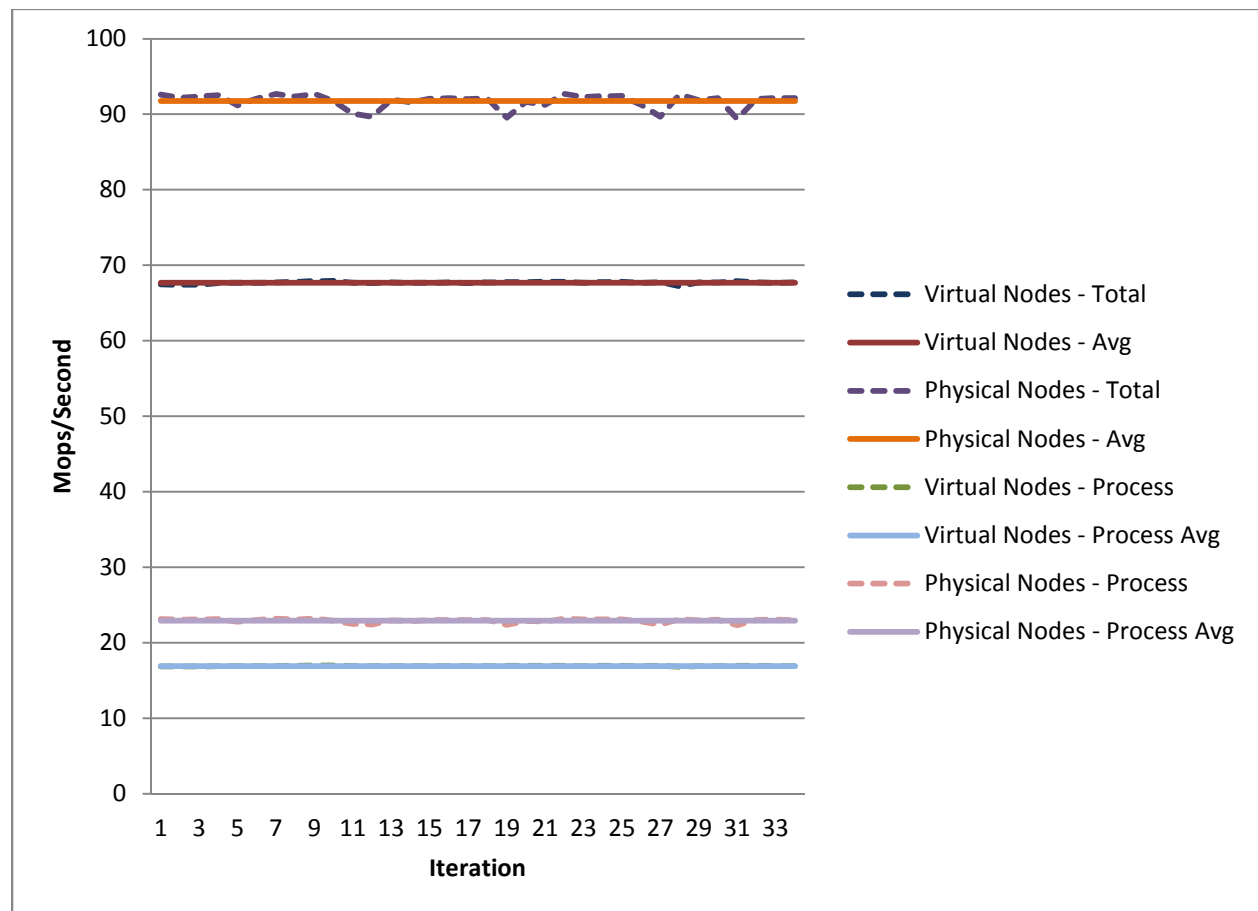
Block Tri-diagonal solver. (NASA 2012)

Figure 13 shows the results of the BT benchmark for both the physical cluster nodes as well as the virtual cluster nodes. In this benchmark the virtual cluster nodes were able to establish a slight advantage over the physical cluster nodes. A closer inspection of the lines reveals that for the algorithm used in this benchmark, the data shows less variation for the

physical cluster nodes. Although more defined in the data for the virtual nodes due to a slight drop in processing around iteration 10, both sets of cluster nodes show a slight upward rise in processing as the iterations progress. This could indicate a processor caching of frequently used data in the benchmark.

## NPB - CG

Figure 14 - NPB CG Benchmark



## Results - NPB CG

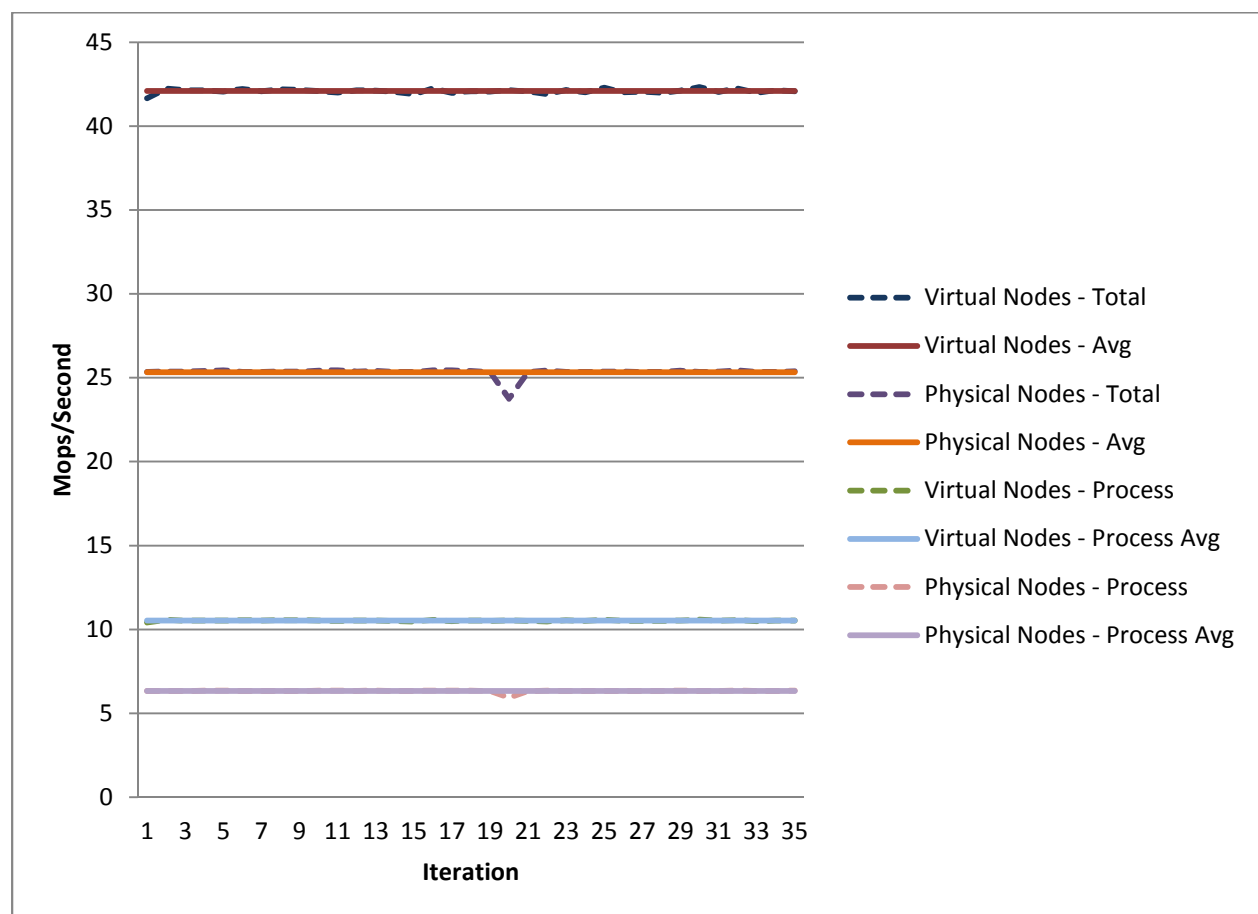
The CG benchmark is a conjugate gradient method used to compute an approximation to the smallest eigenvalue of a large, sparse, symmetric positive definite matrix. This kernel is typical of unstructured grid computations in that it tests irregular long distance communication,

employing unstructured matrix vector multiplication. (Bailey, Barszcz, Barton 1994)

Figure 14 show the results of the CG benchmark. The physical cluster nodes are able to produce a higher benchmarks score. The irregular communication present itself in the graph via a rolling wave shape in the data. This shape is also present in graph for the virtual cluster nodes, though less prevalent. The virtual cluster nodes display less variability in the data as the points are closer to the test average. The data for the physical cluster nodes is the same but with more defined valleys where processing drops. This may be explained by the slower network connection present on the physical nodes.

## NPB - EP

Figure 15 - NPB EP Benchmark



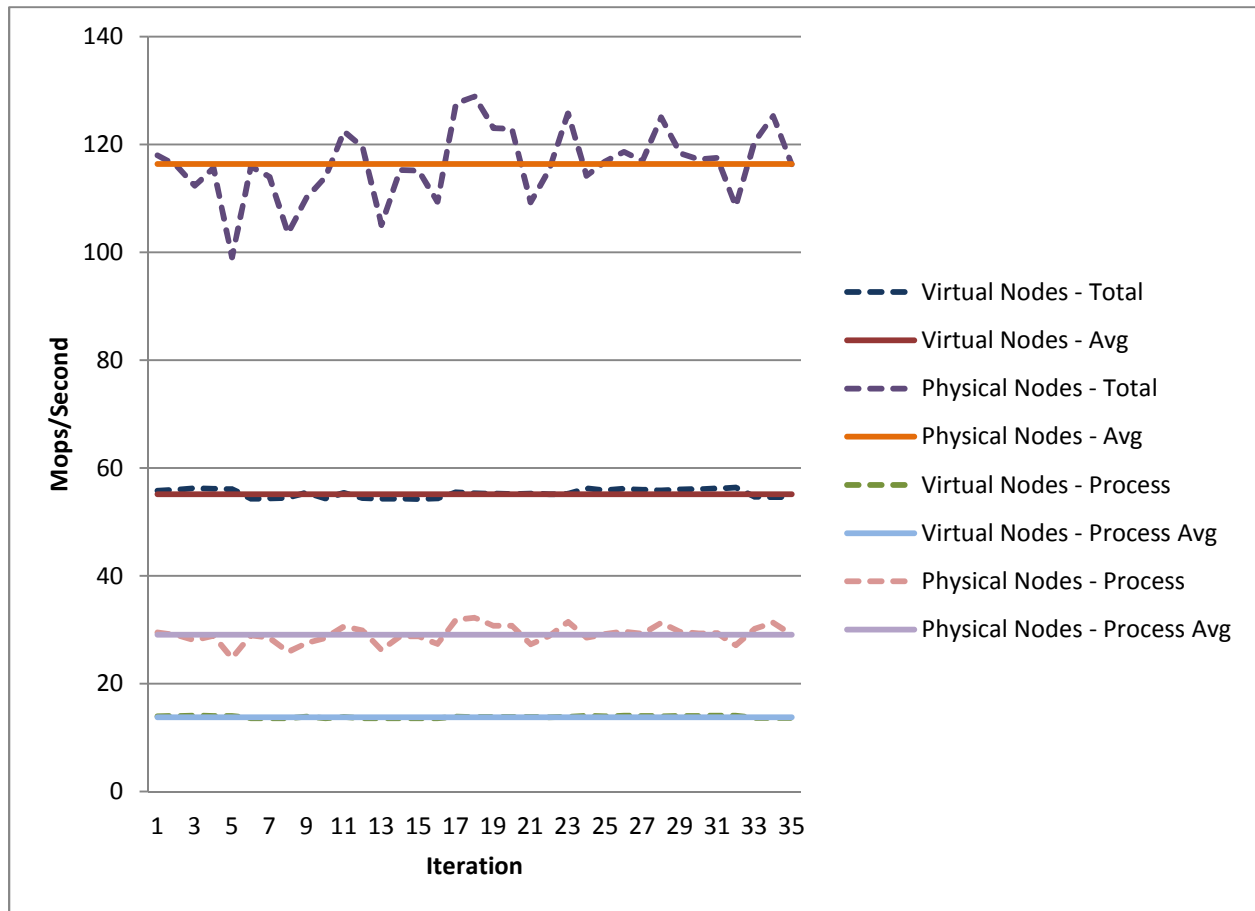
## *Results – NPB EP*

The EP benchmark is the “embarrassingly parallel” kernel. It provides an estimate of the upper achievable limits for floating point performance, i.e. the performance without significant inter-processor communication (Bailey, Barszcz, Barton 1994). This benchmark is named this because there is no inter-node communication required, thus parallel to the number of cluster nodes.

Each virtual cluster node has a 200MHz processor advantage over the physical cluster nodes. This gives the virtual cluster nodes a significant advantage where there is little dependence on inter-node communication. The results are presented in Figure 15. Across thirty-five iterations of this benchmark, the virtual cluster nodes performance nearly doubles that of the physical nodes. For both node types, performance is stable, with little variation from the overall average. The per-process performance is almost indistinguishable from the per-process average.

## NPB – FT

Figure 16 - NPB FT Benchmark



## Results – NPB FT

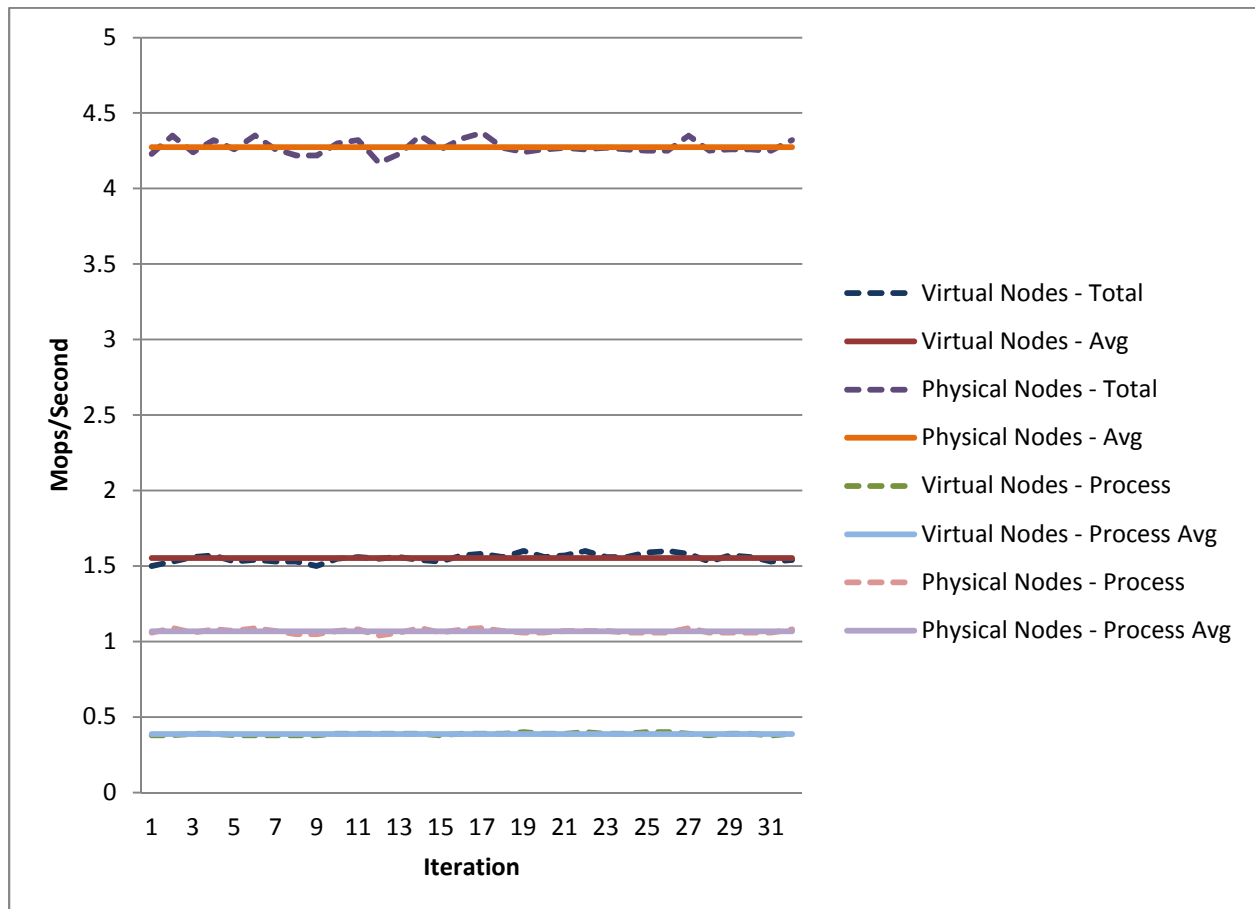
The FT benchmark is a 3-D partial differential equation solution using Fast Fourier Transforms. This kernel performs the essence of many “spectral” codes. It is a rigorous test of long-distance communication performance (Bailey, Barszcz, Barton 1994).

The results of this benchmark are presented in Figure 16. The physical cluster nodes are able to out-perform the virtual cluster nodes. The impact of rigorously testing the long distance communication performance is evident. The physical cluster nodes show a great degree of variability over the entire thirty five iterations. The performance of the virtual cluster nodes

appears to be less impacted in terms of variability, though the performance is roughly half overall.

## NPB - IS

Figure 17 - NPB IS Benchmark



## Results – NPB IS

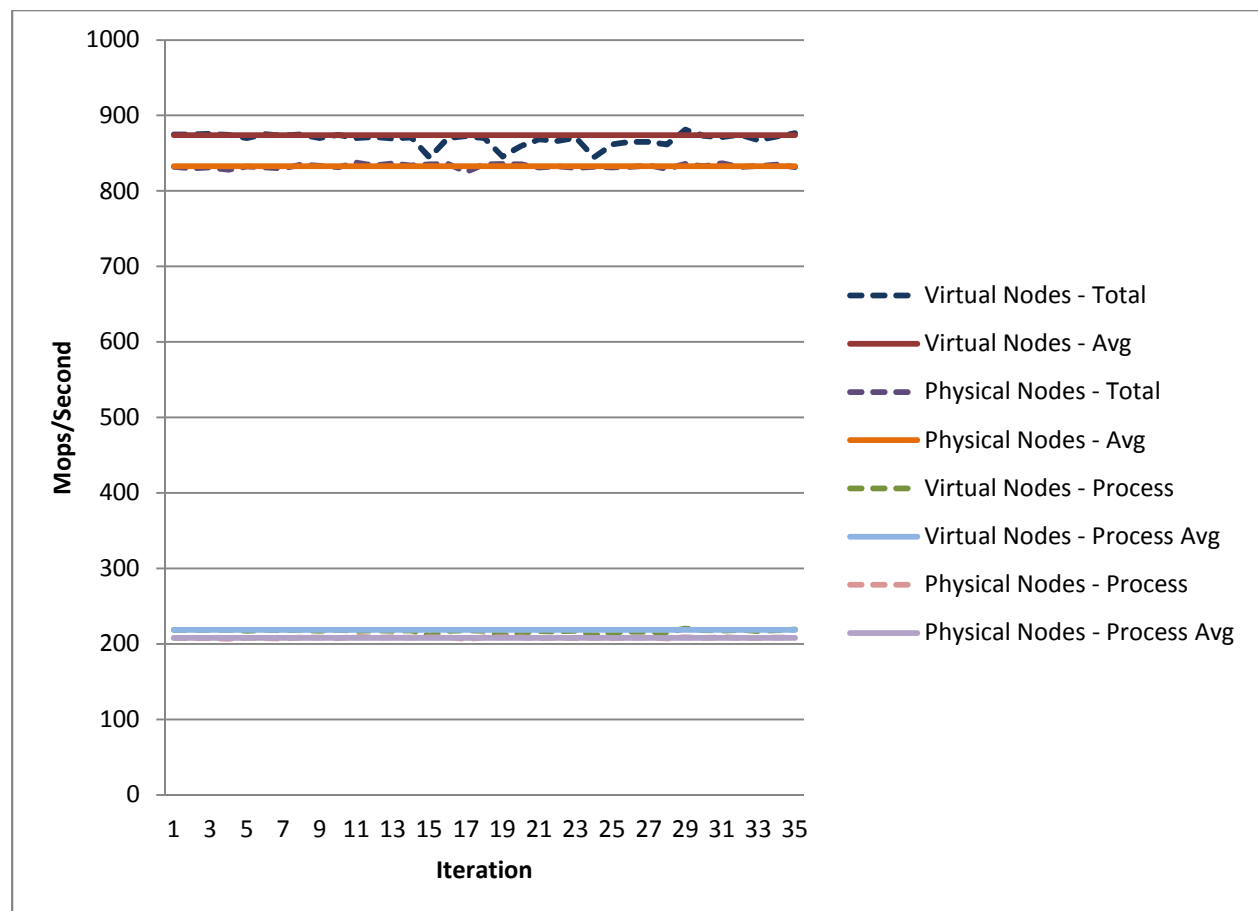
The IS benchmark is a large integer sort. This kernel performs a sorting operation that is important in “particle method” codes. It tests both integer computation speed and communication performance (Bailey, Barszcz, Barton 1994).

Figure 17 shows the results of the IS benchmark. A processor speed advantage by the virtual cluster nodes does not provide an edge as in the embarrassingly parallel benchmark. This

benchmark does use node to node network communication. This is an area of weakness of the virtual cluster nodes. The resulting performance shown is a performance advantage of almost three times by the physical cluster nodes. Both types of nodes display some variability with the physical nodes showing slightly more over the course of thirty iterations. It is possible that this can be attributed to their slower 100Mbps network connection as the virtual nodes also display this behavior, but with smaller peaks and valleys.

## NPB - LU

Figure 18 - NPB LU Benchmark



## Results - NPB LU

The LU benchmark solves a synthetic system of non-linear partial differential equations

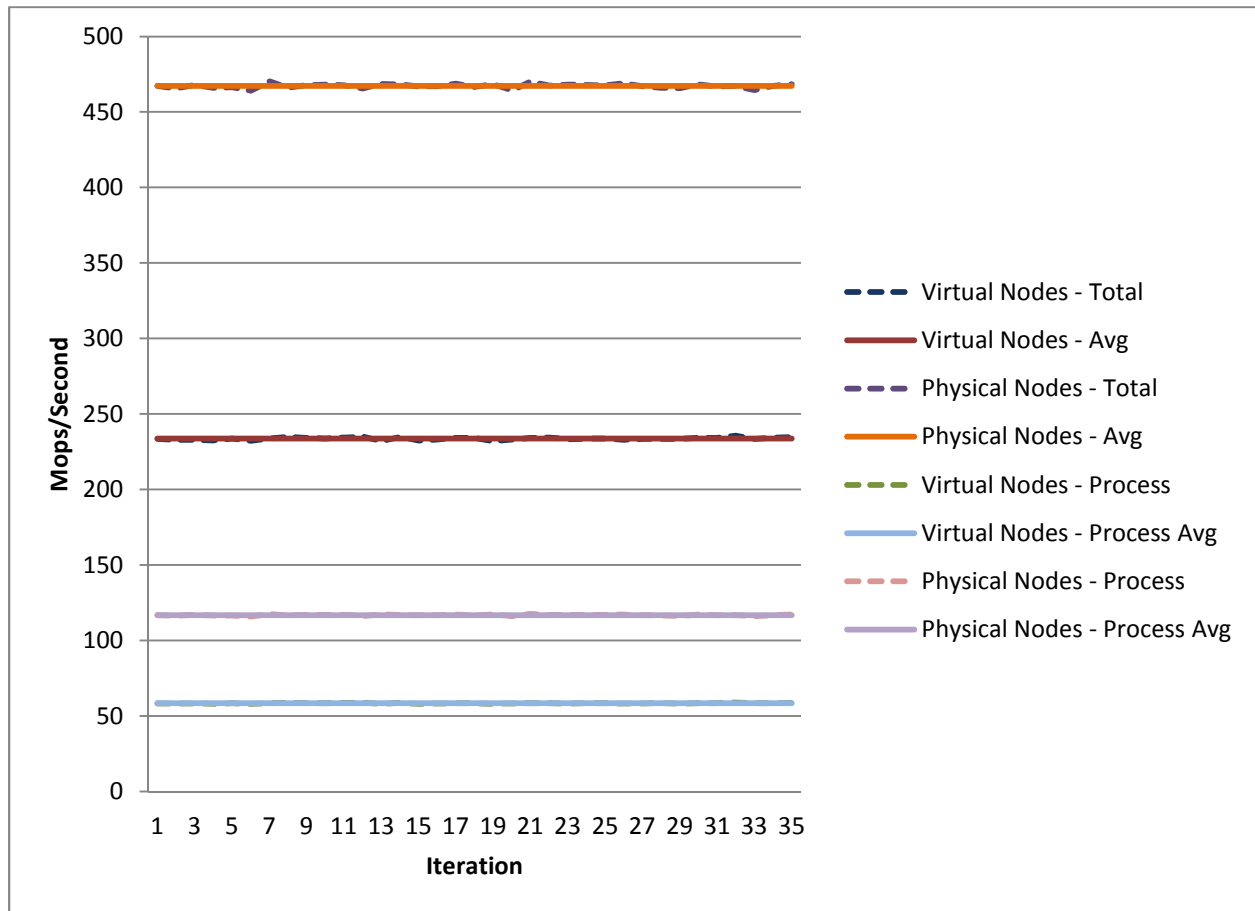


using a Lower-Upper symmetric Gauss-Seidel kernel (NASA 2012).

Figure 18 illustrates the results of the LU benchmark. This was the highest performing benchmark across those utilized in this testing. The virtual cluster nodes were able to obtain nearly 1000 MOPS/Second, followed closely by the physical cluster nodes. The data for the physical nodes closely follow the average while the virtual cluster nodes vary across several iterations. This data shows the performance impact across the virtual cluster for an event that was likely an even happening on the virtualization host itself rather than an even in the benchmark itself. The same pattern of performance drop is not visible on the physical hosts at any point during the thirty five iterations of testing.

## NPB - MG

Figure 19 - NPB MG Benchmark



## Results – NPB MG

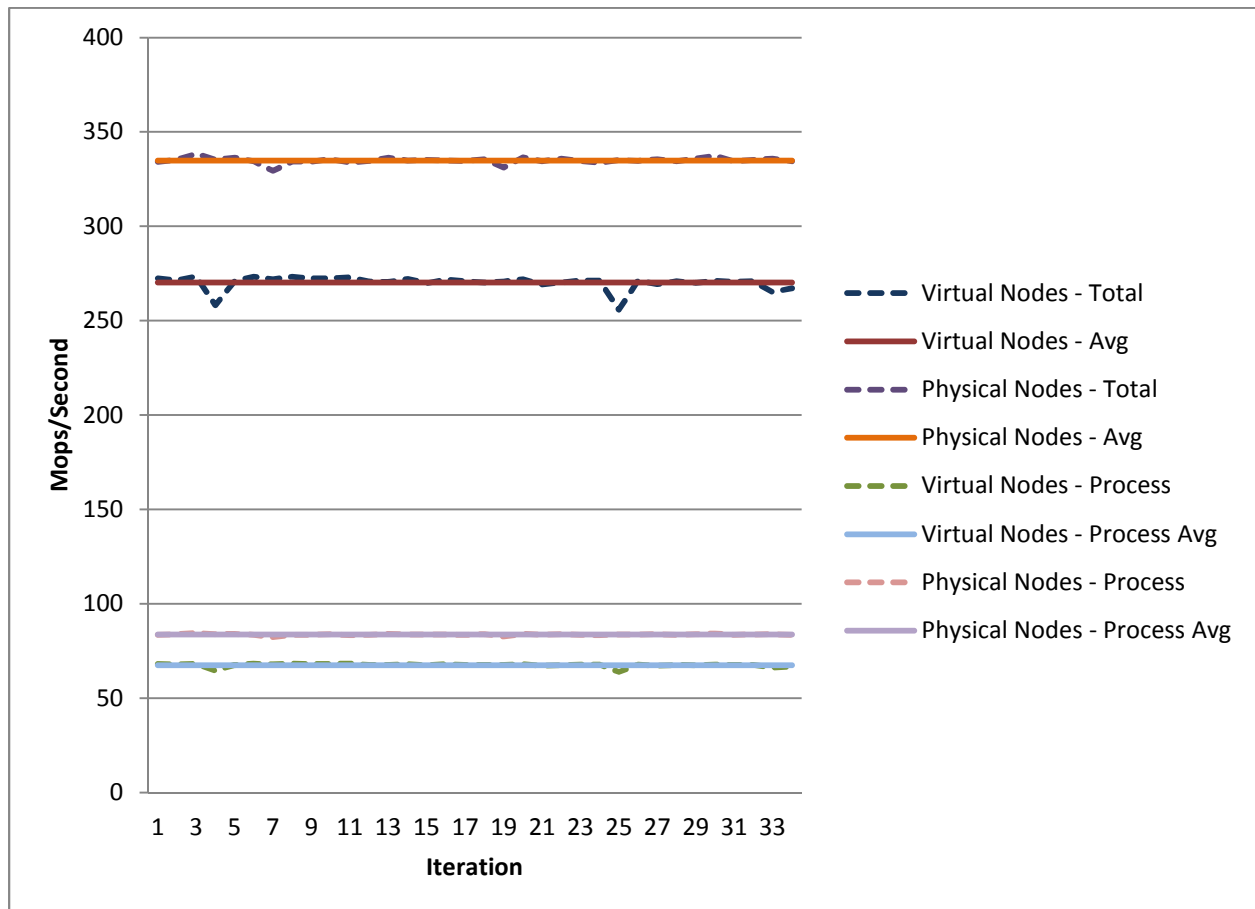
The MG benchmark is Multi-Grid on a sequence of meshes. This benchmark requires high structured long distance communication and tests both short and long distance data communication (Bailey, Barszcz, Barton 1994).

Figure 19 shows that the physical cluster nodes are able to perform better on this benchmark by nearly a factor of two. Performance aside, the virtual cluster nodes and physical cluster nodes data pattern is very similar. There is little variability from the average for either set of nodes over any of the iterations. There are no discernible peaks and valleys as seen in other

NAS benchmarks. The highly structured communication pattern is shown in the presentation of the data.

## NPB - SP

Figure 20 - NPB SP Benchmark



## Results – NPB SP

The SP benchmark is a Scalar Pentadiagonal symmetric successive over-relaxation solver kernel for nonlinear partial differential equations (NASA 2012).

Figure 20 illustrates the results of the SP benchmark. The physical cluster nodes perform better than the virtual cluster nodes and show less variability overall. Three drops in performance pull the average down slightly. These events do not appear to be outlier data as

similar events can be seen in the data on the physical nodes. On the virtual cluster nodes, this could be an even on the virtual host itself, but as these drops in performance can be seen in the data on the physical nodes, it is likely to be a common point in the benchmark, thus impacting both groups.

Figure one through eight displays the results of running the NAS Parallel Benchmarks on the physical and virtual clusters. The highest performance on five of the eight benchmarks was obtained by the physical cluster nodes. Three of these five benchmarks specifically utilize node to node (network) communication to a larger extent. Simultaneous access to the host network resources appears to be an area where virtualization can use improvement. The two highest performing benchmarks, LU and BT, show that access to a higher clock rate processor provides an advantage in operations where little communication is necessary between the cluster nodes. In these two benchmarks, the virtual cluster nodes outperform the physical cluster nodes.

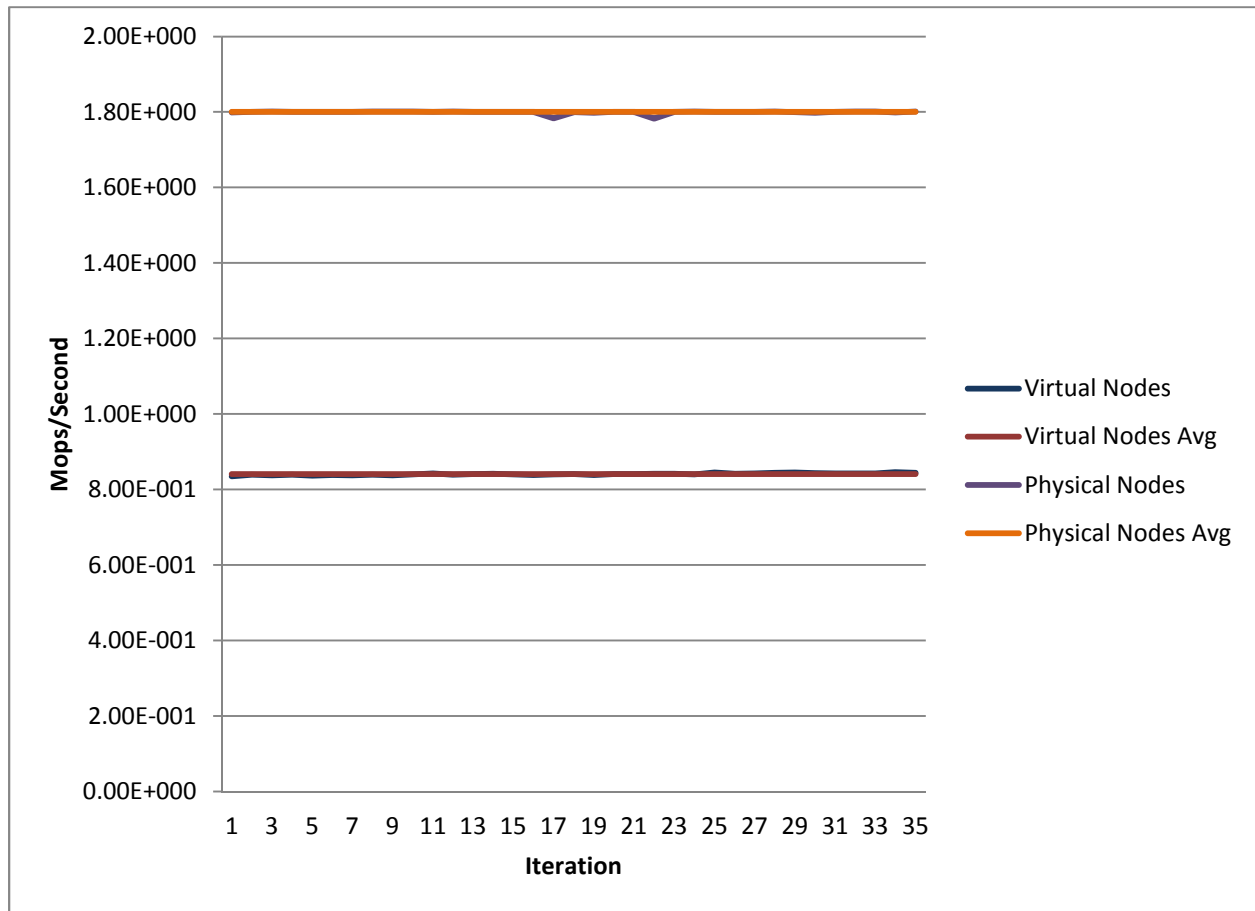
#### ***Benchmark Description – High-Performance Linpack (XHPL)***

“The Linpack Benchmark is a measure of a computer’s floating-point rate of execution. It is determined by running a computer program that solves a dense system of linear equations.”  
(Top500.org)

Unlike the NAS Parallel Benchmarks, XHPL is configurable in order to obtain maximum performance for a given parallel computer as well as to troubleshoot problem areas for new parallel computer installations. Configuration is done via a file named HPL.dat by default. The configuration options used to complete this benchmark can be viewed in Appendix *Figure 31*.

## XHPL

Figure 21 - XHPL Benchmark



## Results - XHPL

Figure 21 displays the results of the XHPL benchmark completed for thirty five iterations. For both the physical cluster nodes and the virtual cluster nodes, the data values show little variation. Both the physical and virtual cluster nodes show a small number of iterations that fall outside the average. The physical virtual nodes show two iterations in a short time-frame that drop below the average. The virtual nodes display the opposite. Three iterations appear above the average, while the single initial iteration falls below. The final approximately eight iterations appear to be above the iteration average as well.

### ***Conclusions, recommendations and financial implications***

Despite the potential of utilizing high performance computer nodes in a virtualized environment, this project has uncovered a number of unexpected drawbacks. The largest of these is the negative impact on performance of simultaneous access to shared resources such as local disks, shared network adapters and parallel/networked file-systems.

The results of the NAS Parallel Benchmarks indicate that I/O is a problem for a virtualized cluster. This is shown in the figures for the CG (p. 34), FT (p. 37), IS (p. 38), MD (p. 41) and SP (p. 42) benchmarks and proceeding results analysis. The NAS Parallel Benchmarks requiring the greatest amount of inter-node communication were the worst performing. For those benchmarks requiring little inter-node communication, the virtual cluster was able to out-perform the physical cluster. The figures for the BT (p. 33), EP (p. 35), and LU (p. 39) benchmarks illustrate this type of performance.

The results of the XHPL benchmark (p. 44) indicate that there is also a disparity between the capabilities of floating point operations on physical cluster nodes versus virtual cluster nodes. A goal of this project was to compare the virtual and physical cluster nodes using the same configuration. Despite having a higher CPU clock rate, the virtual cluster nodes were out-performed by a wide margin using the same configuration file for testing each. With this benchmark, physical cluster nodes produced the highest benchmark scores.

Due to the performance observed in this research, virtualized clusters appear to be a viable option for use in high-performance computing using the components specified for computing with little inter-node communication. Performance characteristics for computations requiring significant inter-node communication should be evaluated carefully prior to deployment in a virtualized environment.

Virtualized clusters do represent a potential flexibility that is more difficult to obtain with physical machines. One such opportunity to utilize this flexibility would be to use virtualized cluster resources on servers during times where utilization is low, such as off-hours and on weekends. In circumstances where top performance is the key driver, dedicated physical cluster nodes are the best solution. In situations where performance can be sacrificed for the option to multi-purpose, virtualized cluster nodes may prove to be an option.

### ***Opportunities for further research***

There are a multitude of changes that could be made to this project for further research. Each benchmark could be tuned for optimum performance rather than focusing on maintaining consistent configuration. Another such change that would make an immediate impact on both the performance and stability of the virtual machines would be to deploy them to a dedicated SAN to eliminate local host disk/file-system issues and to allow the virtual machines to be backed up via SAN snapshots. A second possibility would be to conduct the same benchmarks again using alternate virtualization software on the host machine, such as VMWare's ESX server or the new native Linux kernel implementation KVM. A final variation would be to utilize a high-speed, low latency, interconnect such as Infiniband in order to off-set the performance penalty in the network I/O area of virtualization. This would benefit inter-node communication as well as communication with parallel file-systems should one be used.

## References

- Babcock, Charles. (2007). Virtualization, Original Mainframe Flavor. *Information Week*. February 22, 2007 edition. P 22.
- Bookman, Charles. (2003). *Linux Clustering: Building and Maintaining Linux Clusters*. Indianapolis, IN: New Riders.
- International Data Group. (2011, September). *The Power of Virtualization*. Retrieved March 18, 2012 from <http://www.serverwatch.com/ebooks/37619110/95990/1614220/119068?BLUID=20120325063019465154662>
- IEEE Computing Society. (May, 2005). *Virtual Machine Monitors: Current Technology and Future Trends*. Retrieved Feb 14, 2007 from <http://csdl2.computer.org/persagen/DLAbsToc.jsp?resourcePath=/dl/mags/co/&toc=comp/mags/co/2005/05/r5toc.xml&DOI=10.1109/MC.2005.176>.
- Kiyancilar, Nadir. (2006). *A Virtualization Based Solution for Secure On-Demand Cluster Computing*. Unpublished master's thesis, University of Illinois at Urbana-Champaign.
- USENIX Annual Technical Conference. (2005). *Measuring CPU Overhead for I/O Processing in the Xen Virtual Machine Monitor*. Conference proceedings of the annual conference.
- Vrenios, Alex. (2002). *Linux Cluster Architecture*. Indianapolis, IN: Sams.
- <http://www.vmware.com>
- <http://www.xensource.com>
- <http://www.virtualiron.com>
- Cluster File Systems, Inc. (2007). *Lustre 1.6 Operations Manual*. Boulder, CO: Cluster File



Systems, Inc. Retrieved on 06/13/2007.

Yu, Weikuan & Noronha, Ranjit & Liang, Shuang, & Panda, Dhabaleswar. (2006). Benefits of High Speed Interconnects to Cluster File Systems A Case Study with Lustre. Ohio State University, Department of Computer Science and Engineering.

Huang, Liu, Abali, Panda. (2006). *A Case for High Performance Computing with Virtual Machines*. Ohio State University, Computer Science and Engineering Department and IBM TJ Watson Research Center.

Yu, Weikuan & Vetter, Jeffery. (2008). *Xen-Based HPC: A Parallel I/O Perspective*. Eighth IEEE International Symposium on Cluster Computing and the Grid. Computer Science and Mathematics, Oak Ridge National Laboratories.

Emenecker, Wesley & Jackson, Dave & Butikofer, Joshua & Stanzione, Dan. (2006). *Dynamic Virtual Clustering with Xen and Moab*. Arizona State, Fulton High Performance Computing Institute. Retrieved on 9/15/2007 from

<http://www.springerlink.com/content/uk18j8v37m24824u/>

Cluster File Systems, Inc. (2007). Lustre-Datasheet.

PVFS2 Development Team. (2003). A Quick Start Guide to PVFS2. Parallel Architecture Research Laboratory, Clemson University. Retrieved on 09/15/2007 from <http://www.parl.clemson.edu/pvfs/files.html>.

PVFS2 Development Team. (2003). Parallel Virtual File System, Version 2. Parallel Architecture Research Laboratory, Clemson University. Retrieved on 09/15/2007 from <http://www.parl.clemson.edu/pvfs/files.html>.

Micaela Spigarolo and Renzo Davoli. 2004. Berserkr: a virtual beowulf cluster for fast

prototyping and teaching. In <em>Proceedings of the 1st conference on Computing frontiers</em> (CF '04). ACM, New York, NY, USA, 294-301. DOI=10.1145/977091.977133  
<http://doi.acm.org/10.1145/977091.977133>

NASA Advanced Supercomputing Division. (n.d.) 2012 NAS Parallel Benchmarks. Retrieved March 1, 2012 from <http://www.nas.nasa.gov/publications/npb.html>

<http://www.clusterfs.com/>

<http://www.clustermonkey.net>

<http://cpan.perl.org>

<http://sites.amd.com/us/business/it-solutions/virtualization/Pages/server.aspx>

[http://www.intel.com/technology/virtualization/technology.htm?wapkw=\(Intel+VT\)](http://www.intel.com/technology/virtualization/technology.htm?wapkw=(Intel+VT))

## Appendix A

Figure A1. Lustre Large File Copy

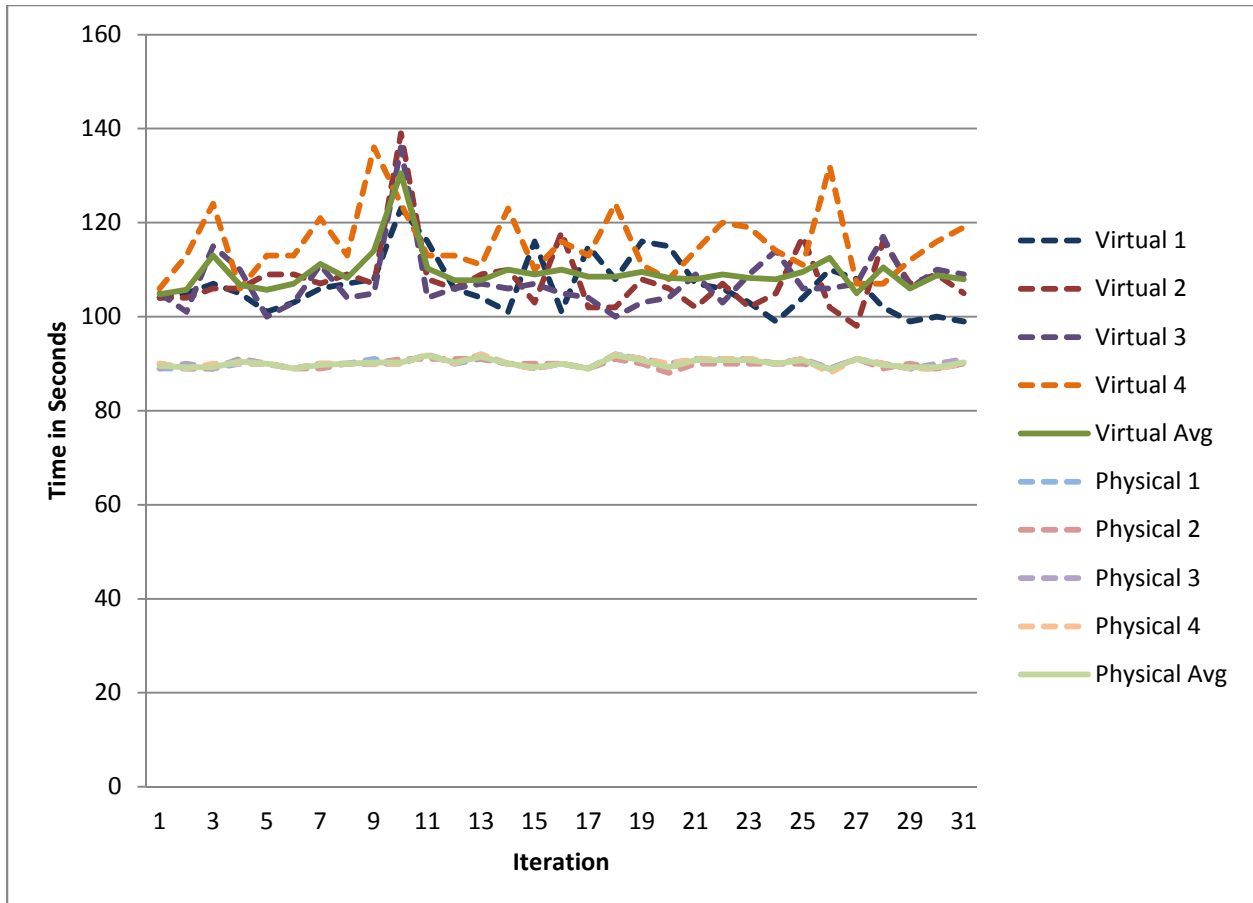


Figure A2. PVFS Large File Copy

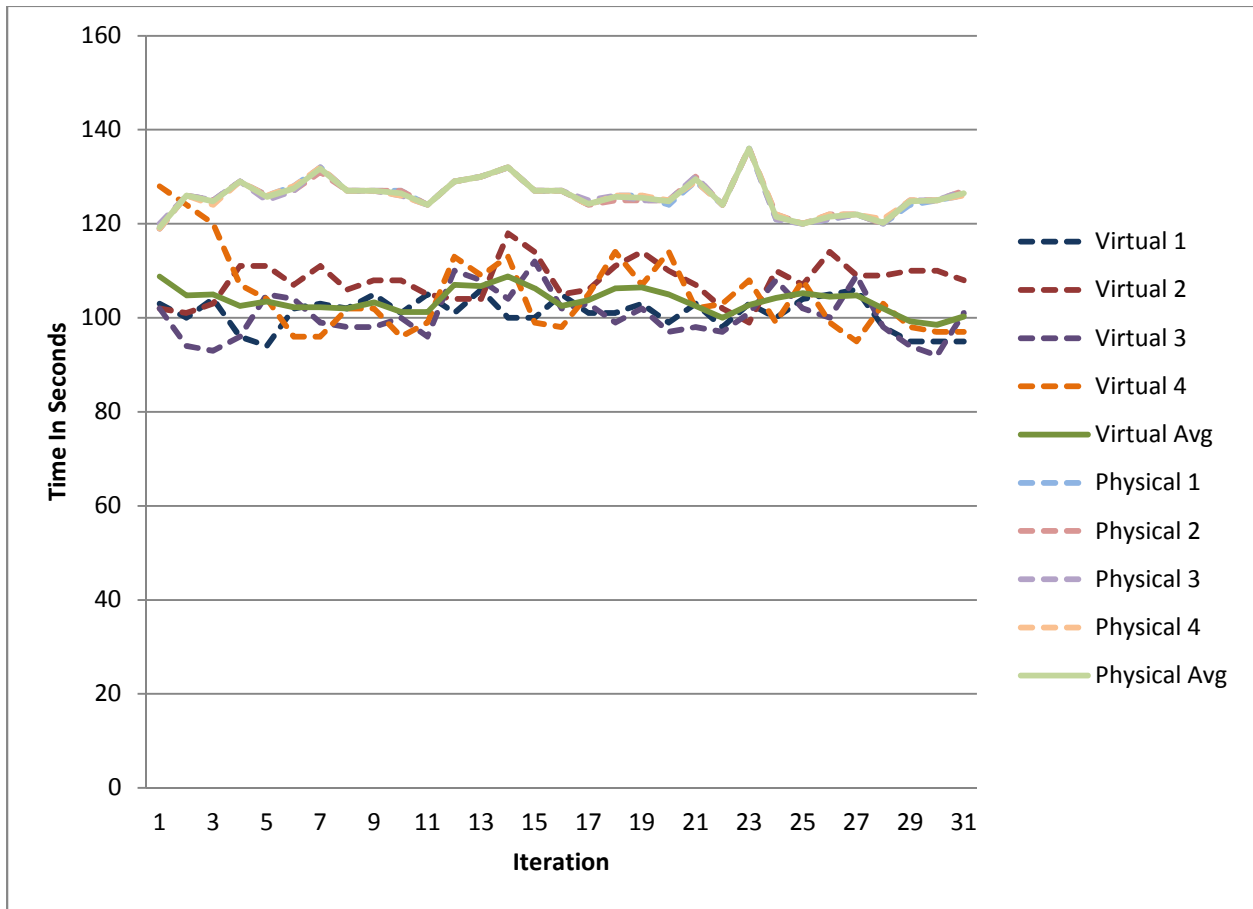


Figure A3. NFS Large File Copy



Figure A4. Averages Large File Copy

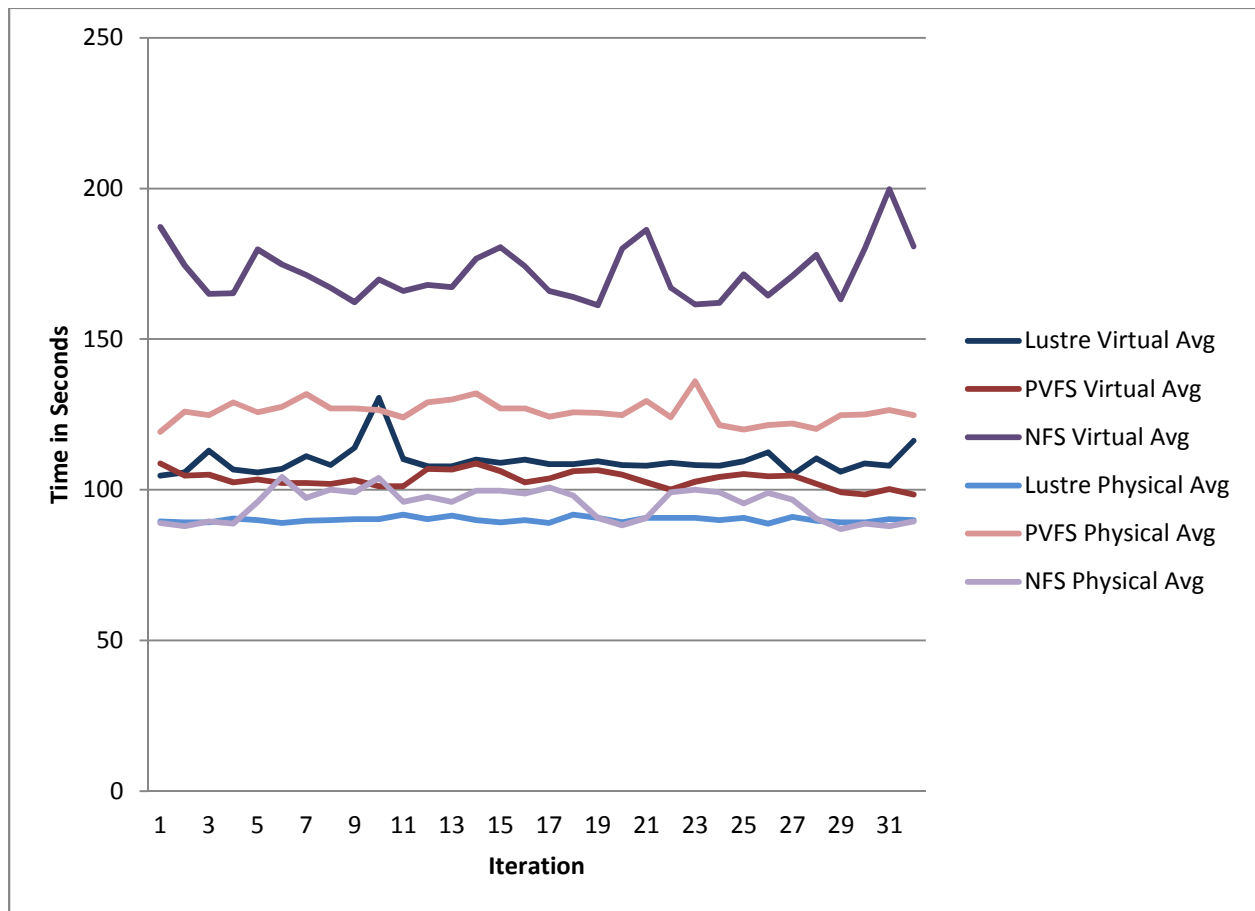


Figure A5. Lustre Small File Copy

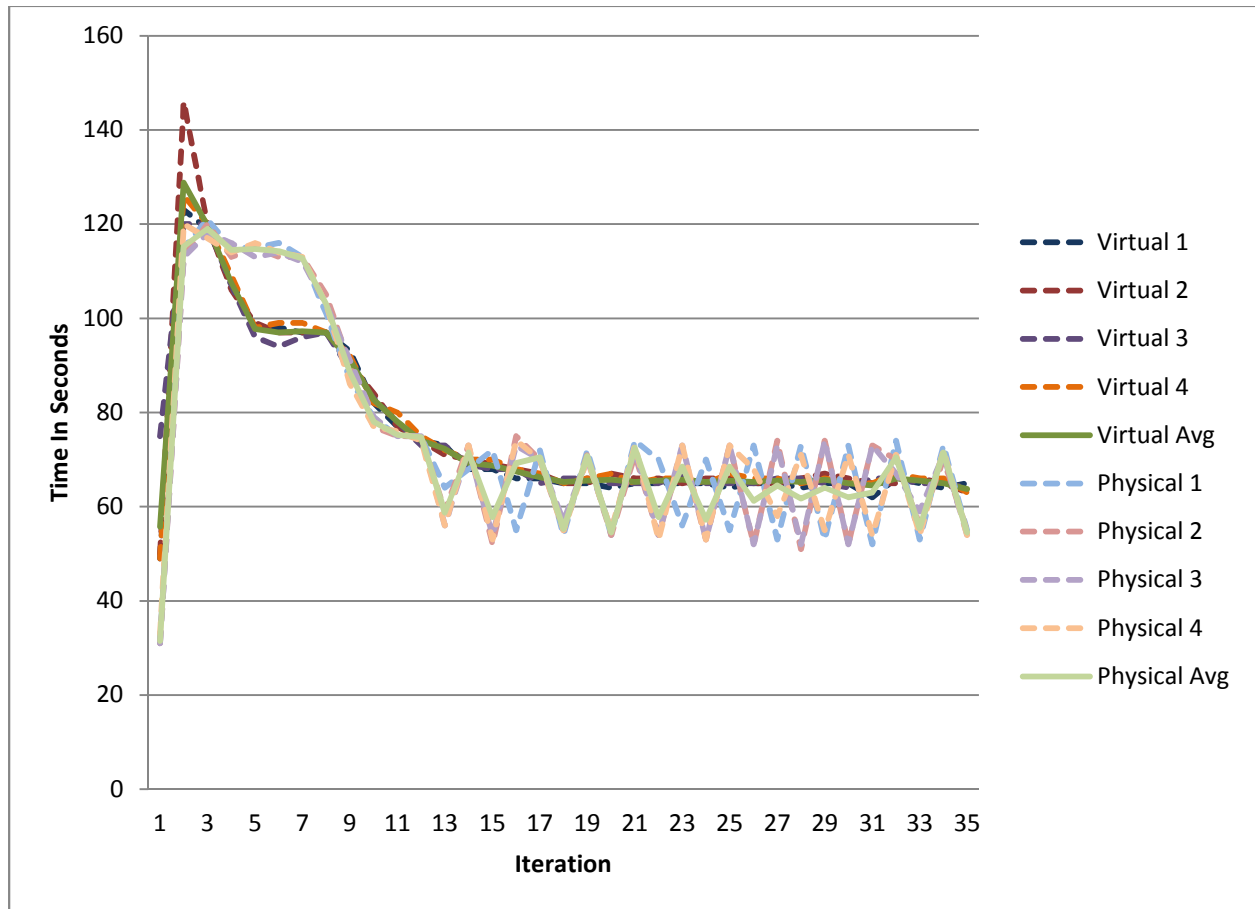


Figure A6. PVFS Small File Copy

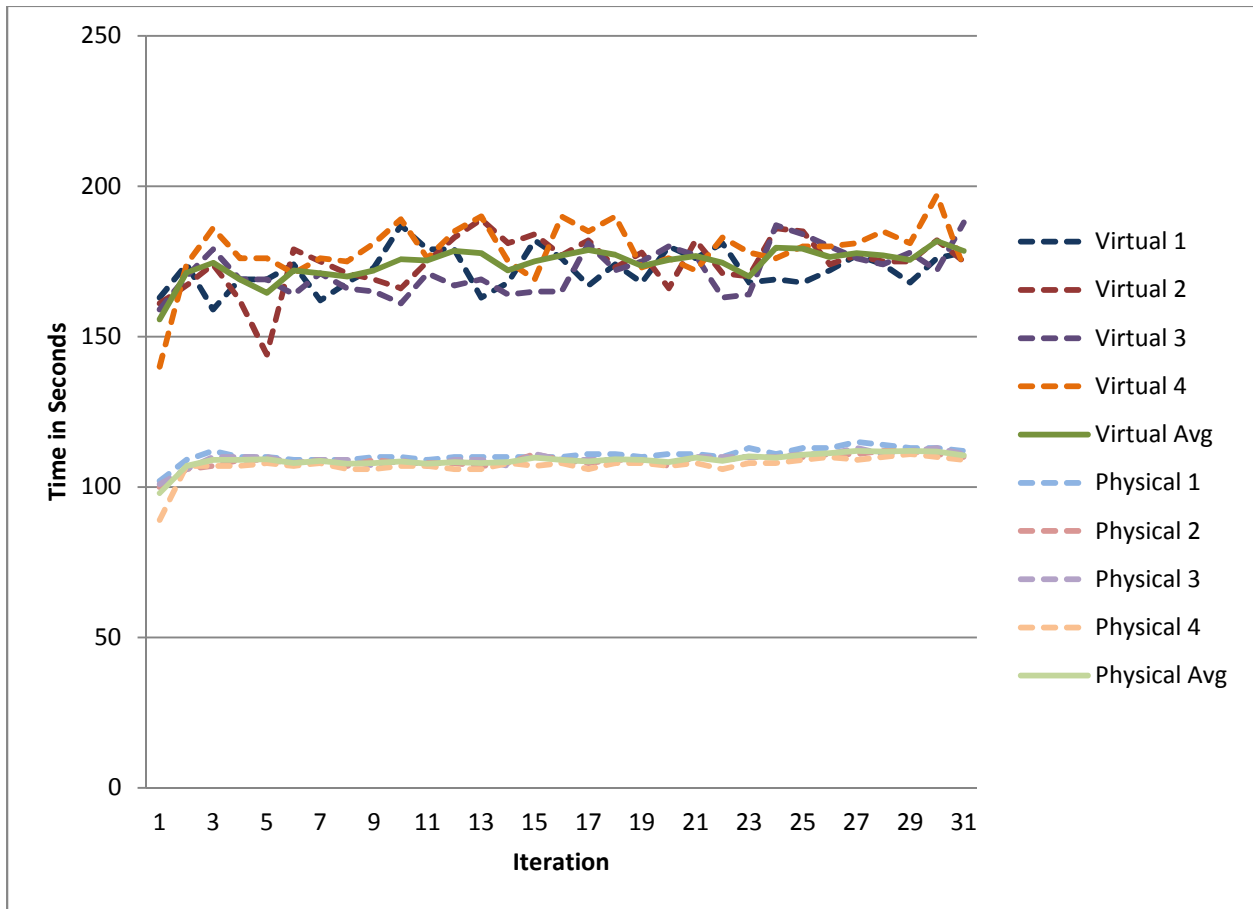




Figure A7. NFS Small File Copy

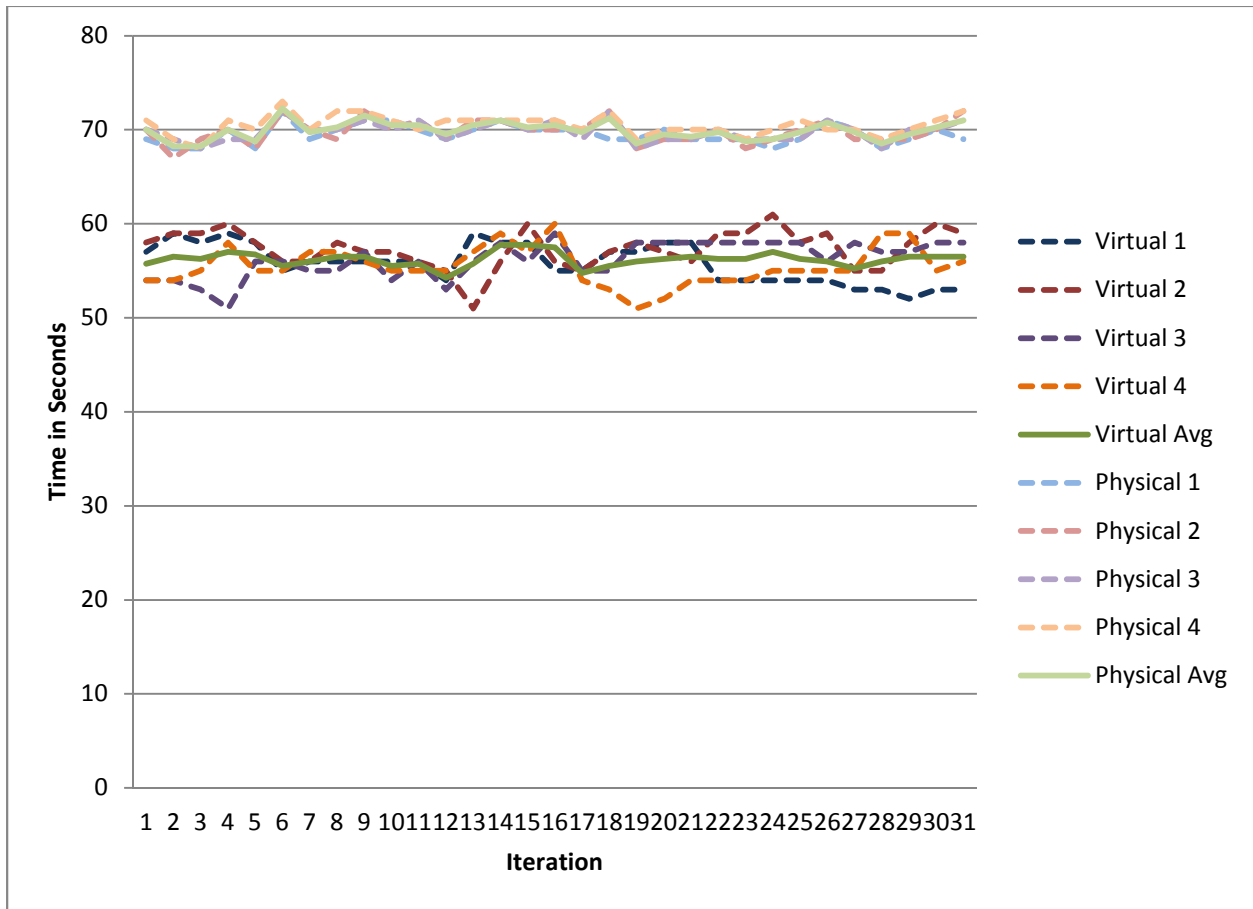


Figure A8. Averages Small File Copy

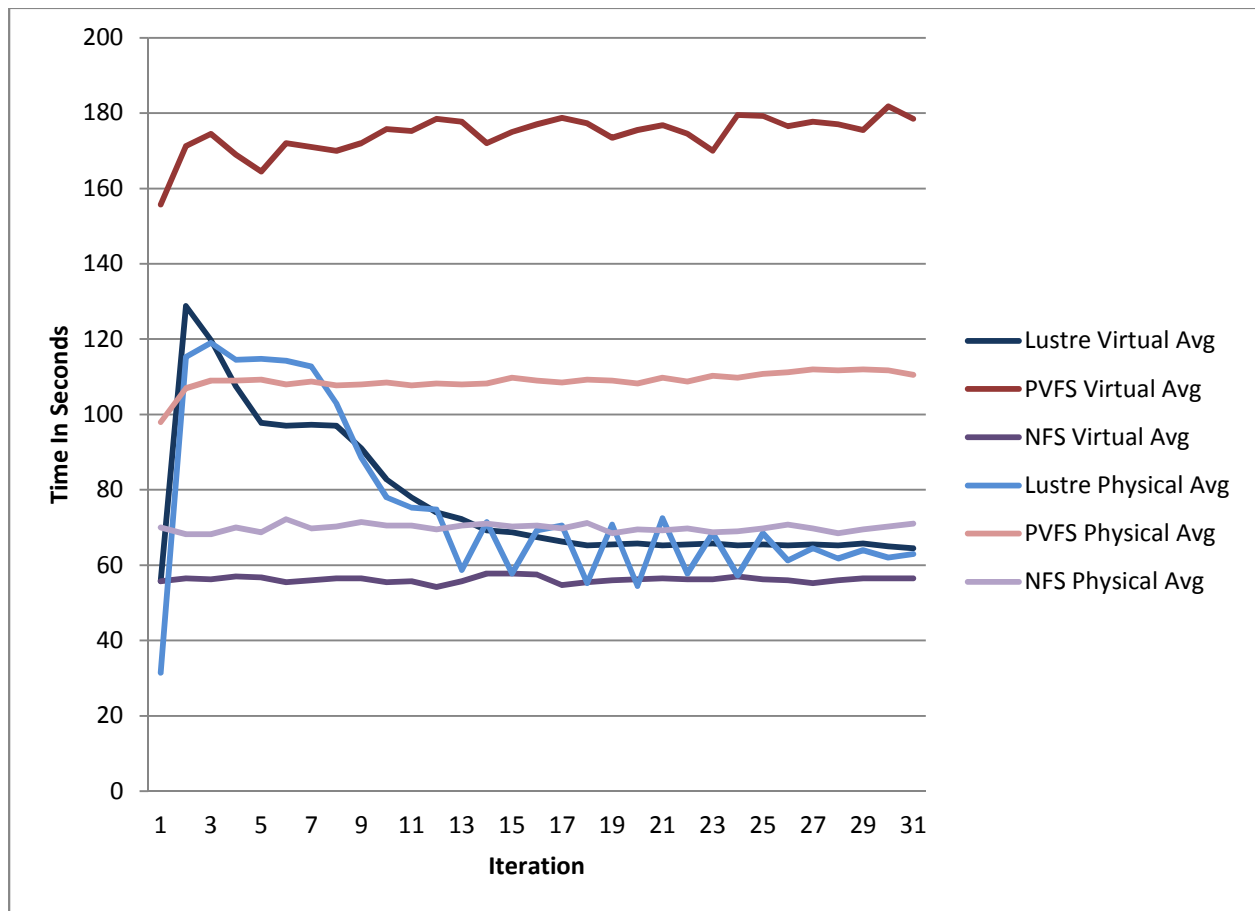


Figure A9. Lustre Small File Writes

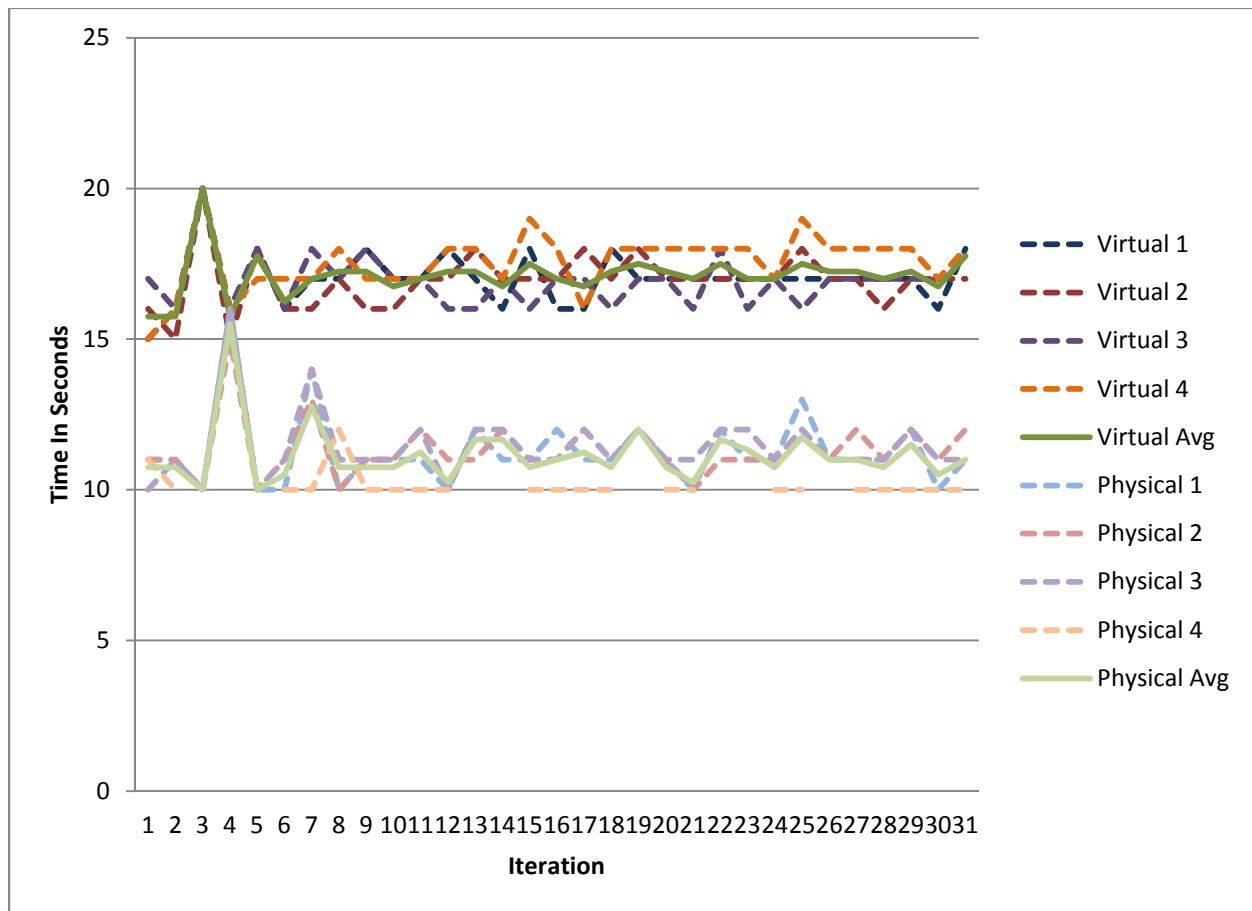


Figure A10. PVFS Small File Writes

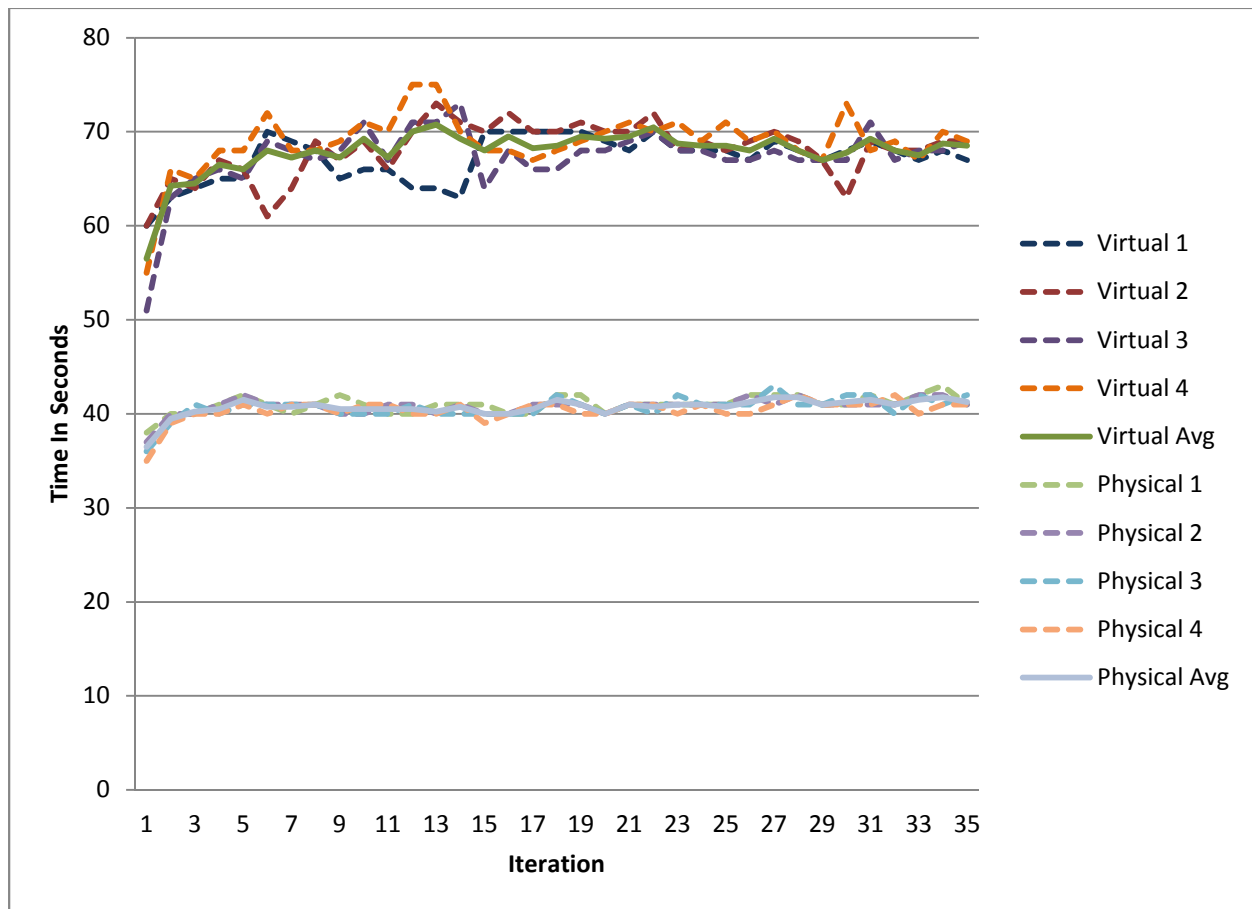


Figure A11. NFS Small File Writes

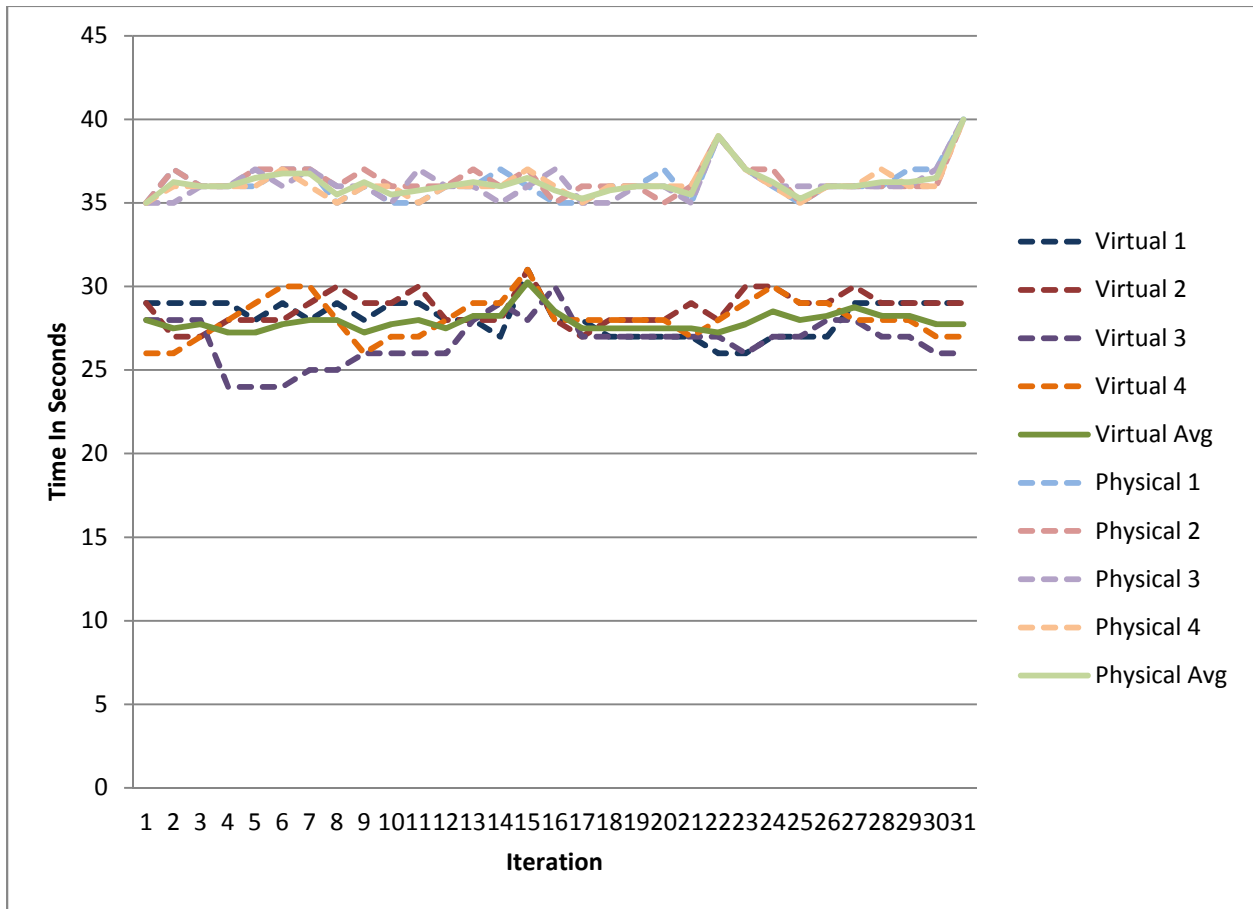


Figure A12. Averages Small File Writes

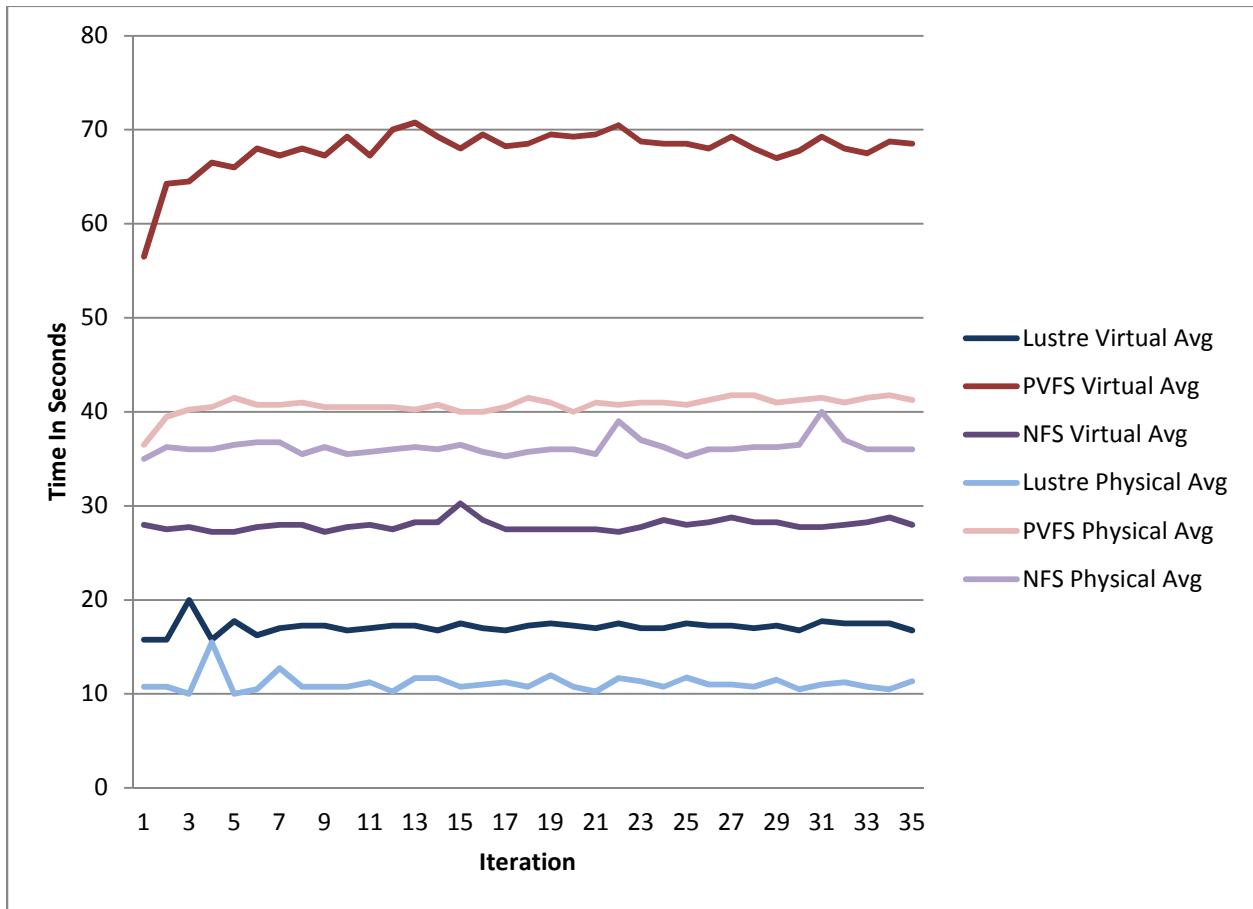


Figure A13. NAS Parallel Benchmarks BT

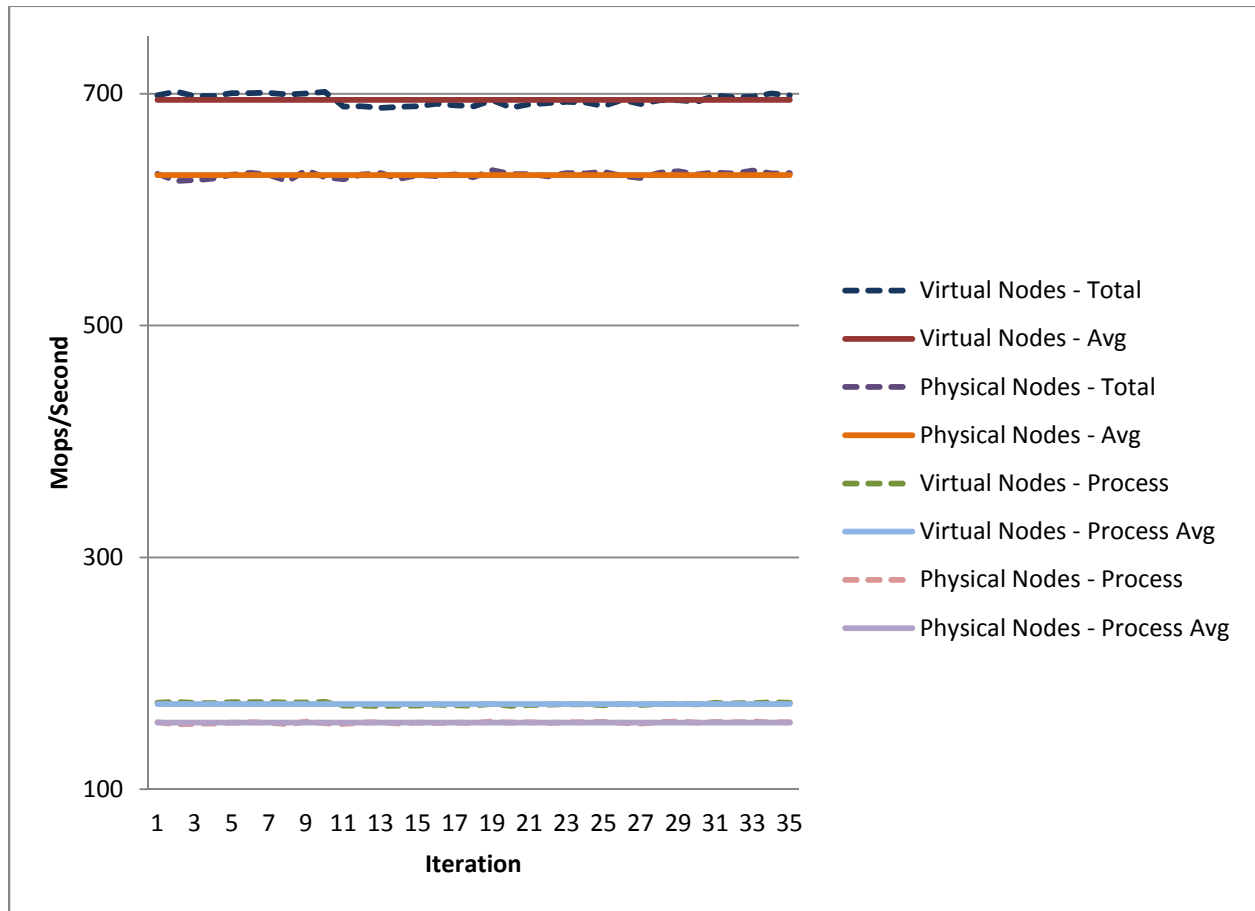


Figure A14. NAS Parallel Benchmarks CG

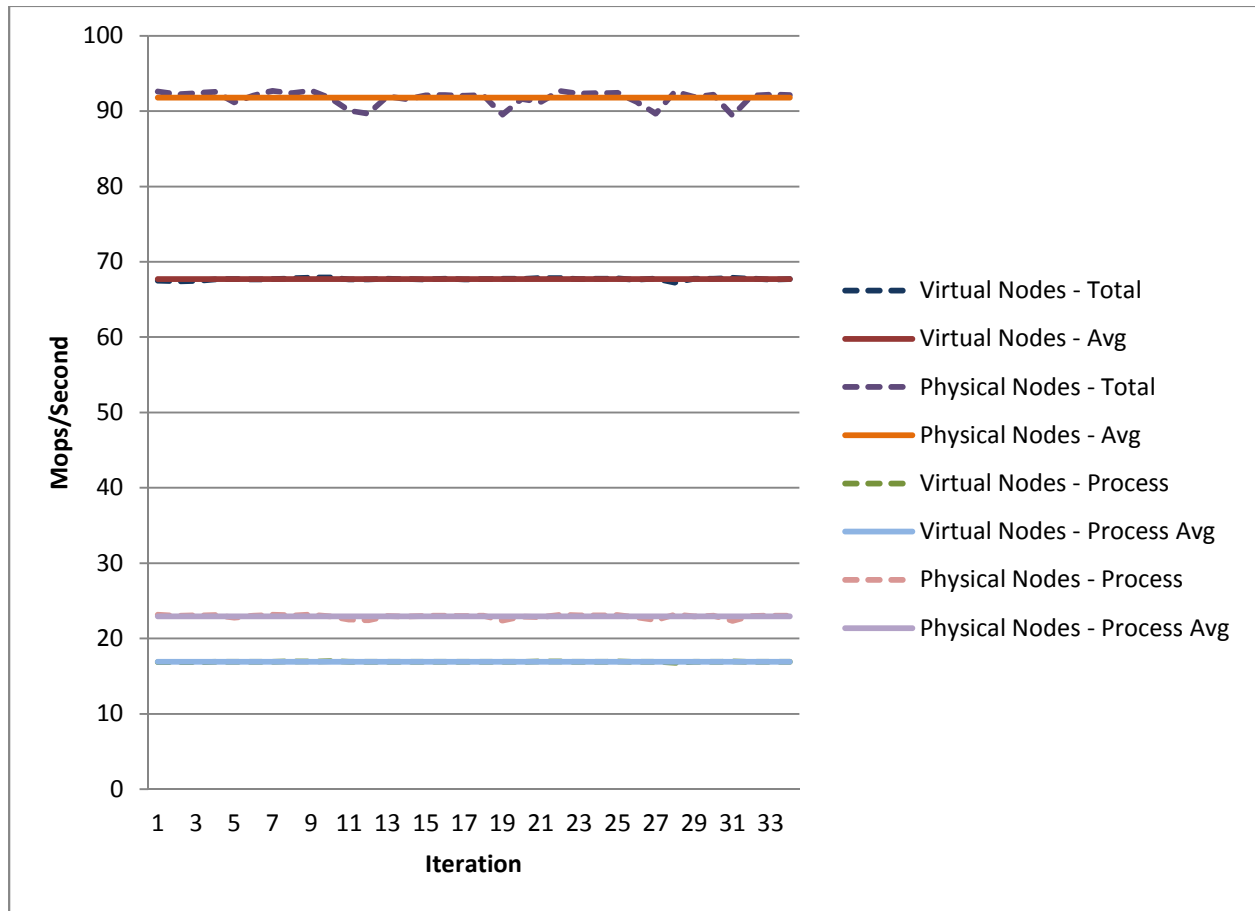




Figure A15. NAS Parallel Benchmarks EP

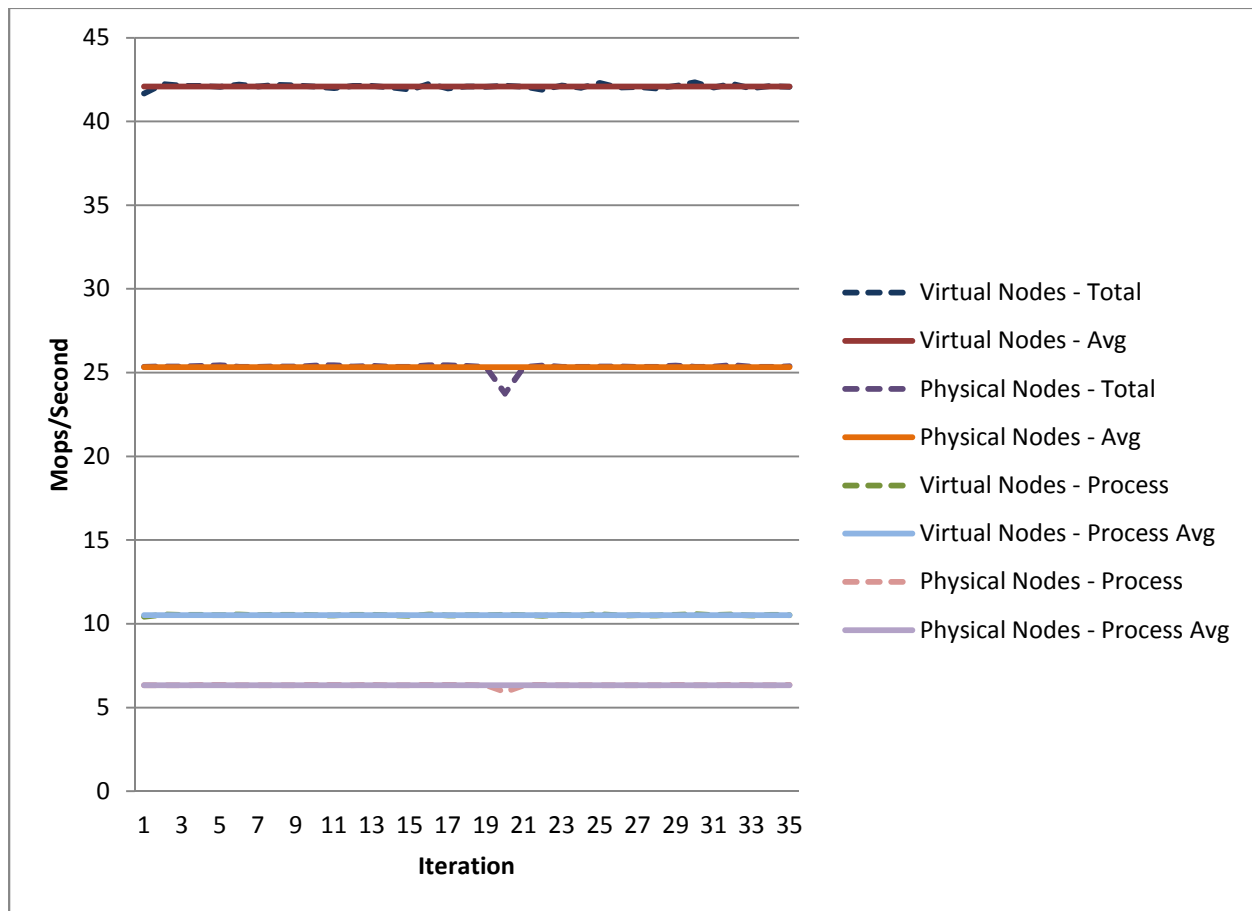


Figure A16. NAS Parallel Benchmarks FT

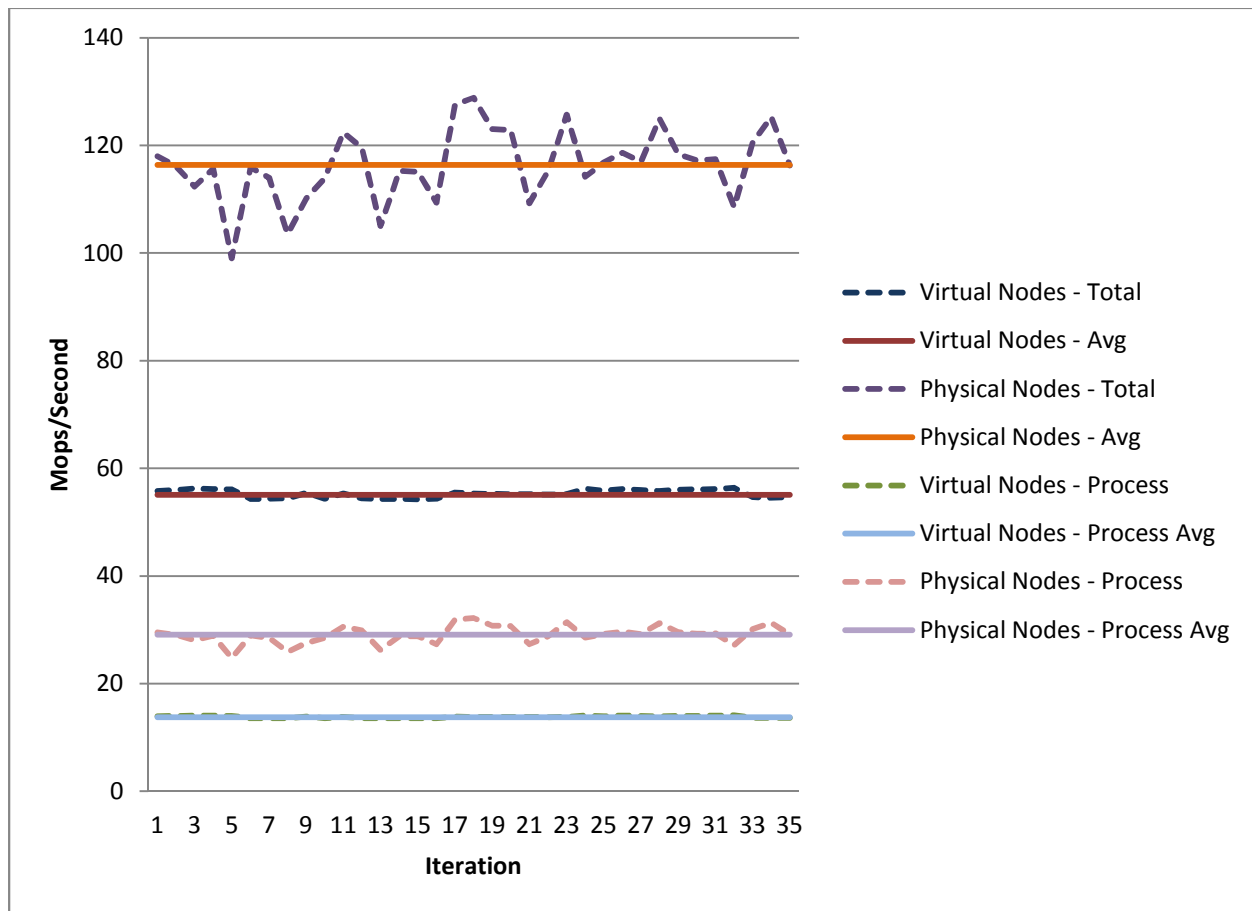


Figure A17. NAS Parallel Benchmarks IS

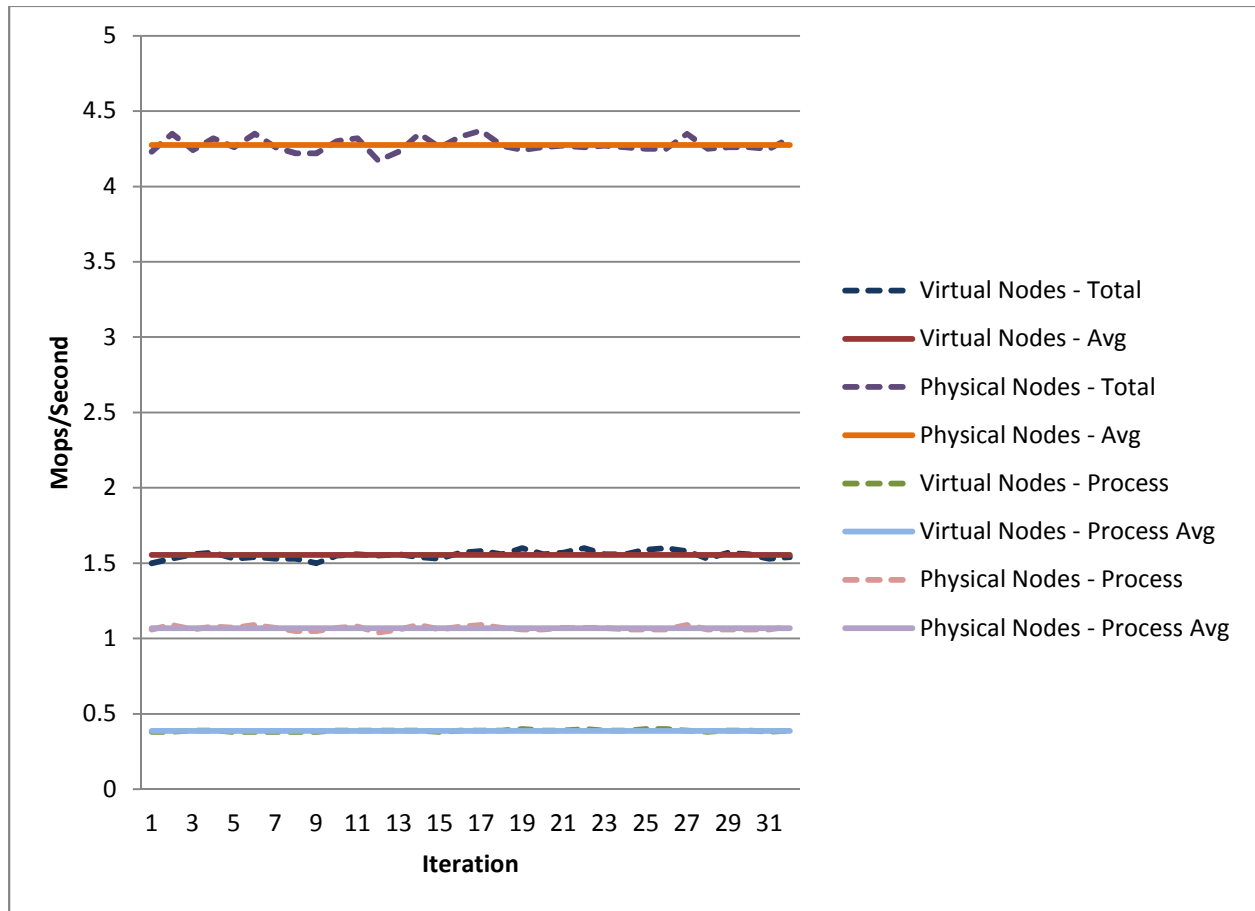


Figure A18. NAS Parallel Benchmark LU

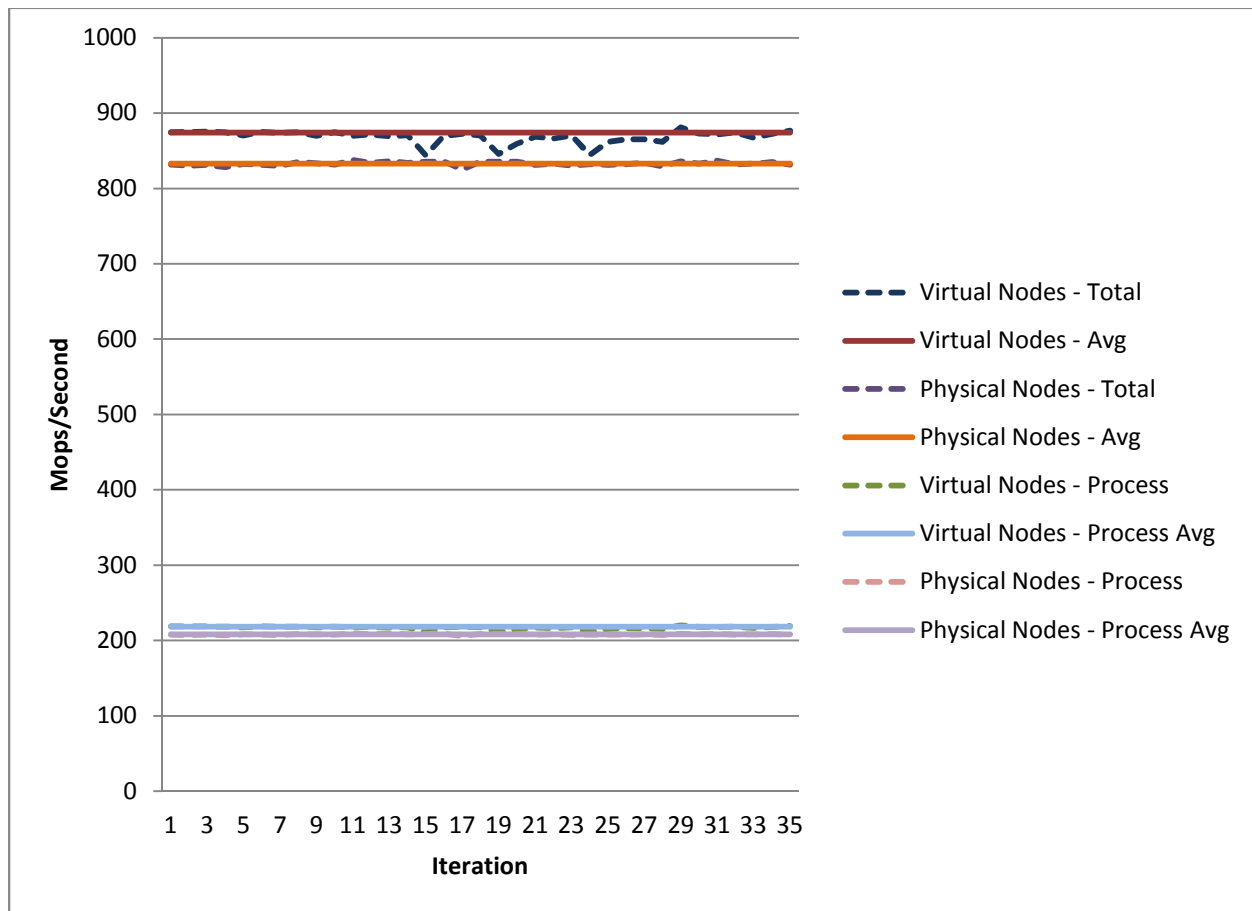


Figure A19. NAS Parallel Benchmarks MG

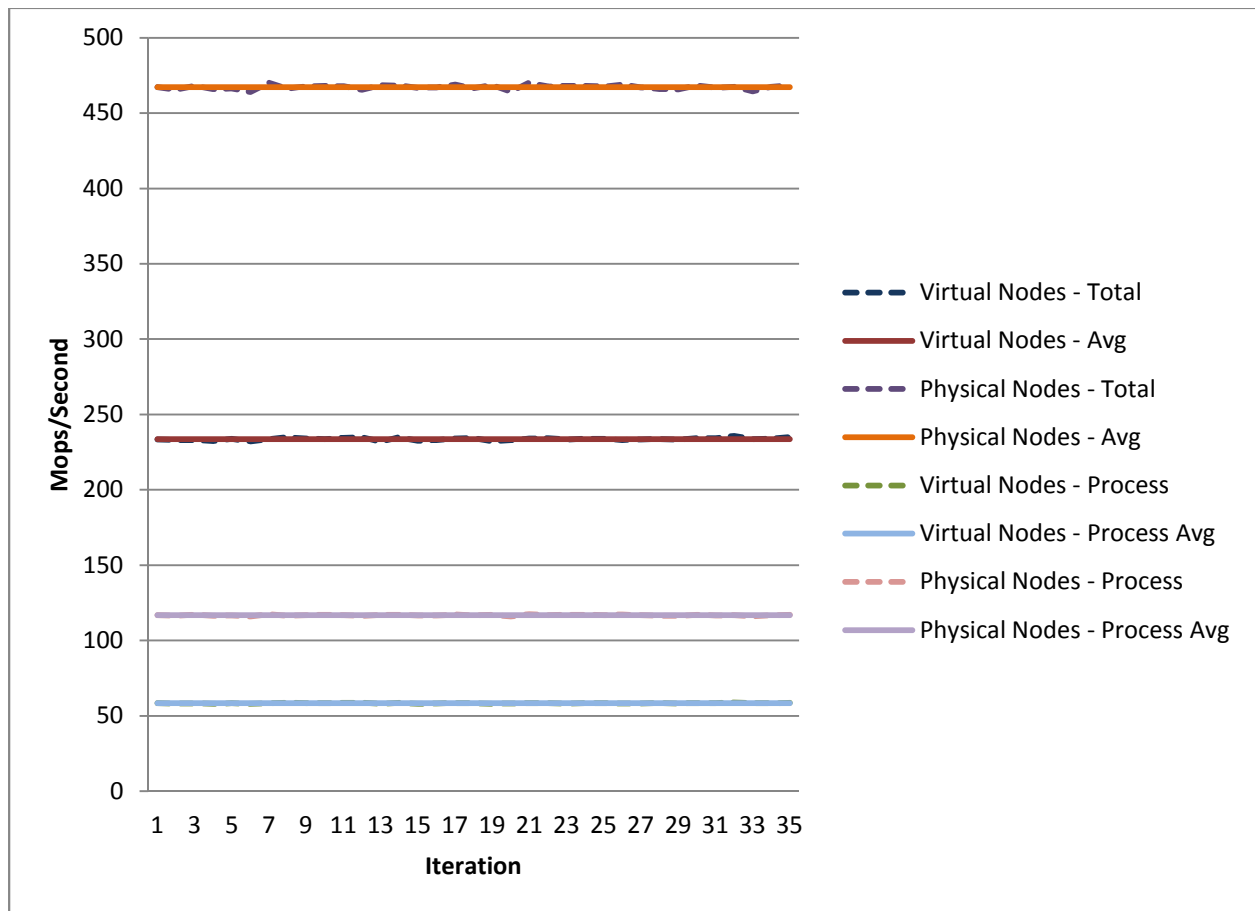


Figure A20. NAS Parallel Benchmarks SP

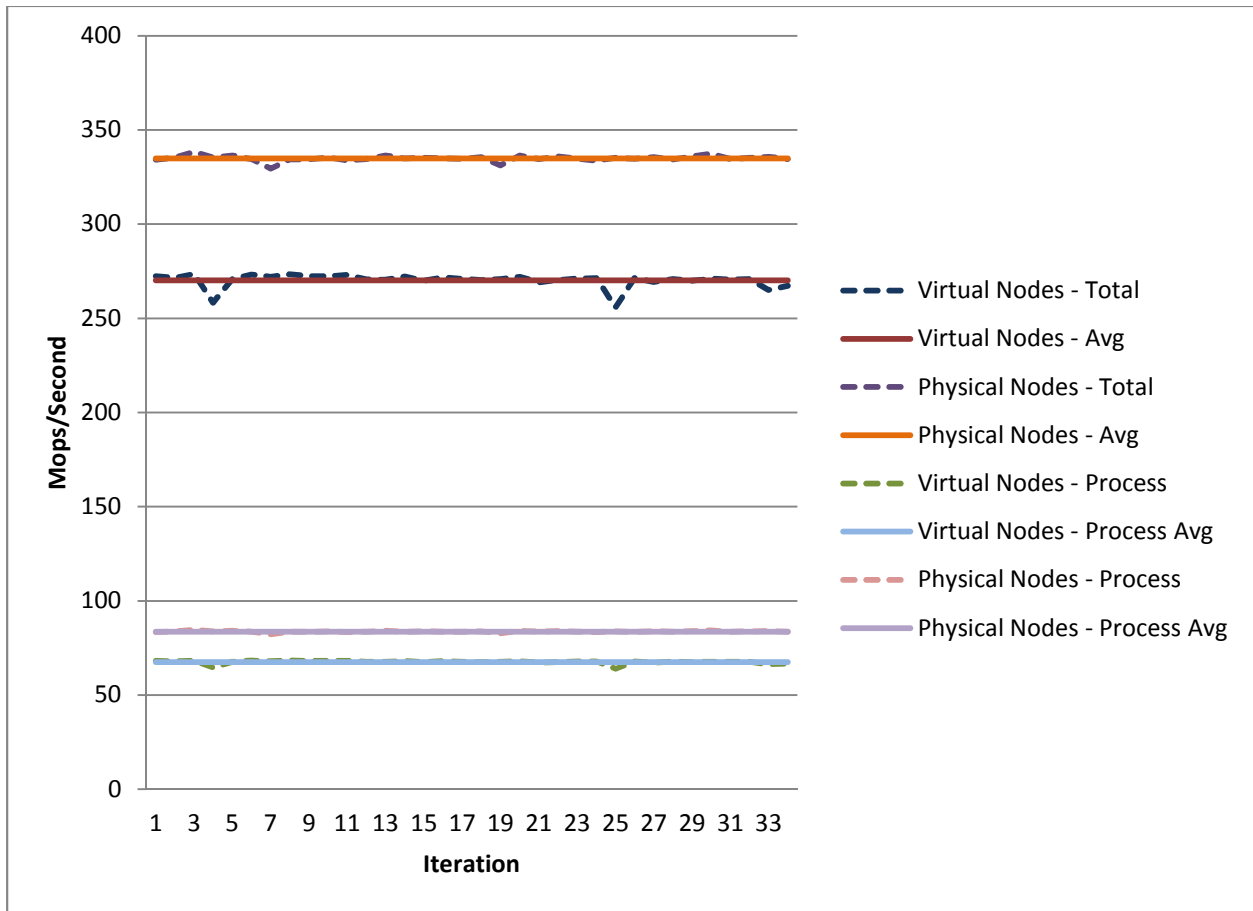


Figure A21. XHPL Benchmarks

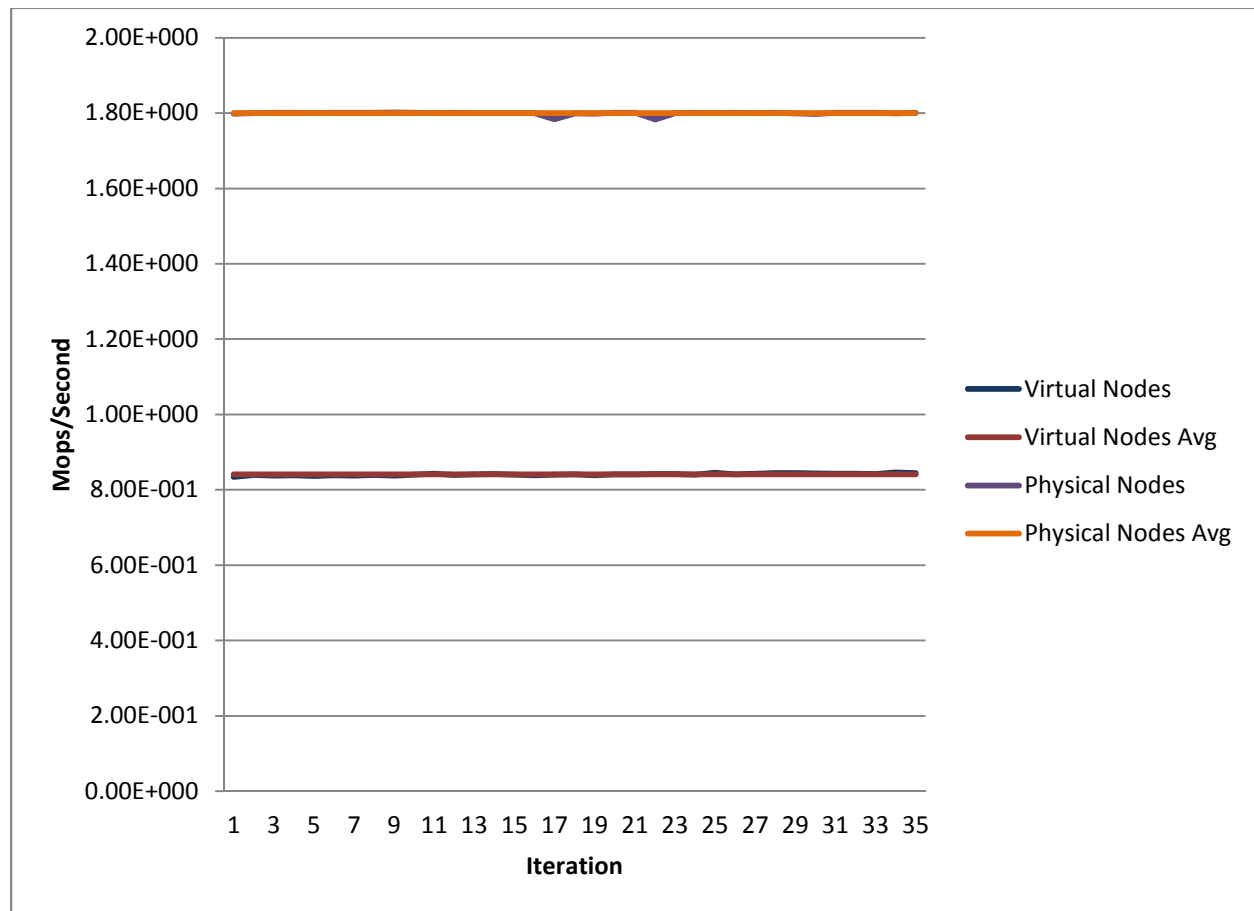


Figure A22. Filecp-lustre.pl

```
#!/usr/bin/perl
#filecp-lustre.pl
use File::Copy;
use Benchmark;
use Socket;
use Sys::Hostname;
open FILE, ">lustre1_4.txt" or die $!;
my $host = hostname();
print FILE $host, "\n";
$source = "/mnt/lustrefs/ubuntu-7.04-server-i386.iso";
$destination = "/tmp/ubuntu-7.04-server-i386.iso";
$starttime = new Benchmark;
$endtime = new Benchmark;
for ($count=0; $count <=35; $count++)
{
    $t0 = new Benchmark;
    copy ($source, $destination) or die "File cannot be copied.";
    $t1 = new Benchmark;
    $td = timediff($t1, $t0);
    ($time,$wallseconds)=split(/\s+/,timestr($td));
    print "$time","\n";
    print FILE $time, "\n";
#    print split(/\s/,timestr($td),$td), "\n";
    unlink($destination);
}
#print "The code took:", $td, "\n";
$totaltime=timediff($endtime,$starttime);
close FILE;
```



Figure A23. Filecp-pvfs.pl

```
#!/usr/bin/perl
#filecp-pvfs.pl
use File::Copy;
use Benchmark;
use Socket;
use Sys::Hostname;
open FILE, ">pvfs1_4.txt" or die $!;
my $host = hostname();
print FILE $host, "\n";
$source = "/mnt/pvfs/ubuntu-7.04-server-i386.iso";
$destination = "/tmp/ubuntu-7.04-server-i386.iso";
$starttime = new Benchmark;
$endtime = new Benchmark;
for ($count=0; $count <=35; $count++)
{
    $t0 = new Benchmark;
    copy ($source, $destination) or die "File cannot be copied.";
    $t1 = new Benchmark;
    $td = timediff($t1, $t0);
    ($time,$wallseconds)=split(/\s+/,timestr($td));
    print "$time", "\n";
    print FILE $time, "\n";
#    print split(/\s+/,timestr($td),$td), "\n";
    unlink($destination);
}
#print "The code took:", $td, "\n";
$totaltime=timediff($endtime,$starttime);
close FILE;
```

Figure A24. Filecp-nfs.pl

```
#!/usr/bin/perl
#filecp-nfs.pl
use File::Copy;
use Benchmark;
use Socket;
use Sys::Hostname;
open FILE, ">nfs1_4.txt" or die $!;
my $host = hostname();
print FILE $host, "\n";
$source = "/mnt/nfsmount/ubuntu-7.04-server-i386.iso";
$destination = "/tmp/ubuntu-7.04-server-i386.iso";
$starttime = new Benchmark;
$endtime = new Benchmark;
for ($count=0; $count <=35; $count++)
{
    $t0 = new Benchmark;
    copy ($source, $destination) or die "File cannot be copied.";
    $t1 = new Benchmark;
    $td = timediff($t1, $t0);
    ($time,$wallseconds)=split(/\s+/,timestr($td));
    print "$time", "\n";
    print FILE $time, "\n";
#    print split(/\s/,timestr($td),$td), "\n";
    unlink($destination);
}
#print "The code took:", $td, "\n";
$totaltime=timediff($endtime,$starttime);
close FILE;
```

Figure A25. Filecp-Lustre\_sf.pl

```
#!/usr/bin/perl
#filecp-lustre_sf.pl
use File::Copy;
use Benchmark;
use Socket;
use Sys::Hostname;

open FILE, ">lustre_sf1_4.txt" or die $!;
my $host = hostname();
print FILE $host,"\n";
for ($outercount=0; $outercount<=34; $outercount++)
{
    $starttime = new Benchmark;
    for ($count=0; $count<=2047; $count++)
    {
        $sourcefilename="/tmp/smallfiles/"."file".".".$count;
        $destinationfilename="/mnt/lustrefs/lustre7/"."file".".".$count;
        print "$sourcefilename";
        print " ";
        print "$destinationfilename";
        system("cp -f $sourcefilename $destinationfilename");
        print "\n";
    }
    $endtime= new Benchmark;
    $td = timediff($endtime,$starttime);
    ($time,$wallseconds)=split(/\s+/,timestr($td));
    print "$time","\n";
    print FILE $time,"\n";
}
close FILE;
```

Figure A26. Filecp-pvfs\_sf.pl

```
#!/usr/bin/perl
#filecp-pvfs_sf.pl
use File::Copy;
use Benchmark;
use Socket;
use Sys::Hostname;

open FILE, ">pvfs_sf1_4.txt" or die $!;
my $host = hostname();
print FILE $host, "\n";
for ($outercount=0; $outercount<=34; $outercount++)
{
    $starttime = new Benchmark;
    for ($count=0; $count<=2047; $count++)
    {
        $sourcefilename="/tmp/smallfiles/" . "file" . "." . $count;
        $destinationfilename="/mnt/pvfs/lustre7/" . "file" . "." . $count;
        print "$sourcefilename";
        print " ";
        print "$destinationfilename";
        system("cp -f $sourcefilename $destinationfilename");
        print "\n";
    }
    $endtime= new Benchmark;
    $td = timediff($endtime,$starttime);
    ($time,$wallseconds)=split(/\s+/,timestr($td));
    print "$time", "\n";
    print FILE $time, "\n";
}
close FILE;
```

Figure A27. Filecp-Lustre\_sf.pl

```
#!/usr/bin/perl
#filecp-nfs_sf.pl
use File::Copy;
use Benchmark;
use Socket;
use Sys::Hostname;

open FILE, ">nfs_sf1_4.txt" or die $!;
my $host = hostname();
print FILE $host, "\n";
for ($outercount=0; $outercount<=34; $outercount++)
{
    $starttime = new Benchmark;
    for ($count=0; $count<=2047; $count++)
    {
        $sourcefilename="/tmp/smallfiles/" . "file" . "." . $count;
        $destinationfilename="/mnt/nfsmount/lustre7/" . "file" . "." . $count;
        print "$sourcefilename";
        print " ";
        print "$destinationfilename";
        system("cp -f $sourcefilename $destinationfilename");
        print "\n";
    }
    $endtime= new Benchmark;
    $td = timediff($endtime,$starttime);
    ($time,$wallseconds)=split(/\s+/,timestr($td));
    print "$time", "\n";
    print FILE $time, "\n";
}
close FILE;
```

Figure A28. Smallfiles-lustre.pl

```
#!/usr/bin/perl
#smallfiles.pl
use File::Copy;
use Benchmark;
use Socket;
use Sys::Hostname;
open FILE, ">sf-lustre1_4.txt" or die $!;
my $host = hostname();
print FILE $host, "\n";
#$delfiles = "/mnt/nfsmount/file*.txt";
for ($outercount=0; $outercount<=63;)
{
    $t0 = new Benchmark;
    for ($count=0; $count<=1023; $count++)
    {
        system("dd if=/dev/urandom of=/mnt/lustrefs/file.$count bs=1024 count=1");
    }
    $t1 = new Benchmark;
    $td = timediff($t1,$t0);
    ($time,$wallseconds)=split(/\s+/,timestr($td));
    print "$time", "\n";
    print FILE $time, "\n";
    select(undef,undef,undef,.250);
    #unlink($delfiles);
    $outercount++;
}
```

Figure A29. Smallfiles-pvfs.pl

```
#!/usr/bin/perl
#smallfiles.pl
use File::Copy;
use Benchmark;
use Socket;
use Sys::Hostname;
open FILE, ">sf-pvfs1_4.txt" or die $!;
my $host = hostname();
print FILE $host, "\n";
#$delfiles = "/mnt/nfsmount/file*.txt";
for ($outercount=0; $outercount<=63;)
{
    $t0 = new Benchmark;
    for ($count=0; $count<=1023; $count++)
    {
        system("dd if=/dev/urandom of=/mnt/pvfs/file.$count bs=1024 count=1");
    }
    $t1 = new Benchmark;
    $td = timediff($t1,$t0);
    ($time,$wallseconds)=split(/\s+/,timestr($td));
    print "$time", "\n";
    print FILE $time, "\n";
    select(undef,undef,undef,.250);
    #unlink($delfiles);
    $outercount++;
}
```

Figure A30. Smallfiles-nfs.pl

```
#!/usr/bin/perl
#smallfiles.pl
use File::Copy;
use Benchmark;
use Socket;
use Sys::Hostname;
open FILE, ">sf-nfs1_4.txt" or die $!;
my $host = hostname();
print FILE $host, "\n";
#$delfiles = "/mnt/nfsmount/file*.txt";
for ($outercount=0; $outercount<=63;)
{
    $t0 = new Benchmark;
    for ($count=0; $count<=1023; $count++)
    {
        system("dd if=/dev/urandom of=/mnt/nfsmount/file.$count bs=1024 count=1");
    }
    $t1 = new Benchmark;
    $td = timediff($t1,$t0);
    ($time,$wallseconds)=split(/\s+/,timestr($td));
    print "$time", "\n";
    print FILE $time, "\n";
    select(undef,undef,undef,.250);
    #unlink($delfiles);
    $outercount++;
}
```



Figure A31. HPLinpack benchmark input file

```

Innovative Computing Laboratory, University of Tennessee
HPL.out    output file name (if any)
6          device out (6=stdout,7=stderr,file)
1          # of problems sizes (N)
4942 Ns
1          # of NBs
16 NBs
0          PMAP process mapping (0=Row-,1=Column-major)
1          # of process grids (P x Q)
1          Ps
4          Qs
16.0       threshold
3          # of panel fact
0 1 2      PFACTs (0=left, 1=Crout, 2=Right)
3          # of recursive stopping criterium
2 4 6      NBMINs (>= 1)
1          # of panels in recursion
2          NDIVs
3          # of recursive panel fact.
0 1 2      RFACTs (0=left, 1=Crout, 2=Right)
1          # of broadcast
0          BCASTs (0=1rg,1=1rM,2=2rg,3=2rM,4=Lng,5=LnM)
1          # of lookahead depth
0          DEPTHs (>=0)
2          SWAP (0=bin-exch,1=long,2=mix)
64         swapping threshold
0          L1 in (0=transposed,1=no-transposed) form
0          U  in (0=transposed,1=no-transposed) form
1          Equilibration (0=no,1=yes)
8          memory alignment in double (> 0)

```

*Figure A32.* Benchmarks.sh/Benchmarks-virt.sh

```
#!/bin/bash
echo "cg"
/home/glen/./submit-cg.pl

echo "mg"
/home/glen/./submit-mg.pl

echo "is"
/home/glen/./submit-is.pl

echo "ep"
/home/glen/./submit-ep.pl

echo "lu"
/home/glen/./submit-lu.pl

echo "ft"
/home/glen/./submit-ft.pl

echo "sp"
/home/glen/./submit-sp.pl

echo "bt"
/home/glen/./submit-bt.pl

echo "xhpl"
/home/glen/./submit-xhpl.pl
```

*Figure A33.* Script-Benchmark.sh

```
#!/bin/bash
#PBS -N bt.B.4
#nodes=4:ppn=1
#PBS -l
nodes=oscarnode1.home.net+oscarnode2.home.net+oscarnode3.home.net+oscarnode4.home.net
#PBS -l walltime=02:00:00
#PBS -r n
# -e stderr
# -o stdout
#PBS -V

echo $PBS_NODEFILE
cat $PBS_NODEFILE

#mpiexec /mnt/lustrefs/homes/glen/ep.A.2 -n 2
#/opt/mpich-1.2.7p1/bin/mpirun /mnt/lustrefs/homes/glen/ep.B.2 -v -machinefile
$PBS_NODEFILE -np 2
/opt/mpich-ch_p4-gcc-1.2.7/bin/mpirun -np 4 -machinefile $PBS_NODEFILE /home/glen/bt.B.4
```

*Figure A34.* Submit-Benchmark.pl

```
#!/usr/bin/perl
use File::Copy;
use Benchmark;
use Socket;
use Sys::Hostname;
```

```
for ($count=1;$count<=35;$count++)
{
    print $count;
    system("qsub script-cg");
    sleep (650);
}
```

## Appendix B

### File-System Benchmarks for physical cluster nodes

*Figure B1.* Lustre Large File Copy (1 of 1)

oscarnode1.home.net

45  
44  
44  
45  
44  
45  
45  
44  
44  
45  
45  
44  
45  
45  
44  
44  
44  
44  
44  
45  
45  
44  
44  
44  
45  
45  
44  
45  
45  
44  
44  
44

*Figure B2.* Lustre Large File Copy (1 of 2)  
oscarnode1.home.net

52  
51  
52  
52  
52  
51  
52  
51  
51  
52  
52  
51  
52  
52  
51  
54  
51  
52  
53  
51  
50  
51  
51  
50  
51  
51  
51  
51  
53  
53  
51  
51  
51  
51  
53  
51

*Figure B3.* Lustre Large File Copy (2 of 2)  
oscarnode2.home.net

53  
51  
52  
52  
52  
51  
51  
52  
51  
52  
52  
51  
52  
51  
51  
51  
53  
51  
52  
51  
51  
51  
51  
50  
50  
52  
51  
52  
54  
53  
52  
51  
51  
52  
51  
51

*Figure B4.* Lustre Large File Copy (1 of 3)

oscarnode1.home.net

69  
69  
69  
69  
69  
70  
68  
69  
68  
70  
69  
69  
68  
68  
70  
71  
68  
69  
70  
70  
70  
69  
69  
69  
70  
69  
70  
69  
69  
69  
70  
69  
69



*Figure B5.* Lustre Large File Copy (2 of 3)  
oscarnode2.home.net

69  
69  
69  
69  
69  
69  
69  
70  
69  
69  
69  
69  
68  
69  
70  
71  
69  
69  
69  
70  
70  
68  
69  
68  
71  
69  
71  
70  
70  
70  
69  
69  
70  
68  
69

*Figure B6.* Lustre Large File Copy (3 of 3)  
oscarnode3.home.net

69  
70  
69  
69  
69  
69  
69  
69  
69  
69  
69  
70  
68  
70  
71  
68  
69  
69  
70  
71  
69  
69  
70  
70  
70  
70  
70  
70  
69  
70  
69  
69  
69  
69

*Figure B7.* Lustre Large File Copy (3 of 3)  
oscarnode3.home.net

69  
70  
69  
69  
69  
69  
69  
69  
69  
69  
69  
70  
68  
70  
71  
68  
69  
69  
70  
71  
69  
69  
70  
70  
70  
70  
70  
70  
69  
70  
69  
69  
69  
69

*Figure B8.* Lustre Large File Copy (1 of 4)  
oscarnode1.home.net

89  
89  
89  
90  
90  
89  
90  
90  
91  
90  
92  
90  
91  
90  
89  
90  
89  
92  
91  
89  
91  
91  
91  
90  
91  
89  
91  
90  
89  
89  
90  
90  
91  
91  
90

*Figure B9.* Lustre Large File Copy (2 of 4)  
oscarnode2.home.net

90  
89  
89  
91  
90  
89  
89  
90  
90  
91  
91  
91  
91  
90  
90  
90  
89  
91  
90  
88  
90  
90  
90  
90  
90  
89  
91  
89  
90  
89  
90  
90  
90  
92  
90  
91

*Figure B10.* Lustre Large File Copy (3 of 4)  
oscarnode3.home.net

89  
90  
89  
91  
90  
89  
90  
90  
90  
90  
92  
90  
92  
90  
89  
90  
89  
92  
91  
90  
91  
91  
91  
90  
91  
89  
91  
90  
89  
90  
91  
90  
91  
91  
90

*Figure B11.* Lustre Large File Copy (4 of 4)  
oscarnode4.home.net

90  
89  
90  
90  
90  
89  
90  
90  
90  
90  
92  
90  
92  
90  
89  
90  
89  
92  
91  
90  
91  
91  
91  
90  
91  
88  
91  
90  
89  
89  
90  
90  
91  
91  
91

*Figure B12.* NFS Large File Copy (1 of 1)  
oscarnode1.home.net

47  
47  
48  
47  
47  
47  
47  
47  
50  
47  
47  
49  
47  
48  
47  
48  
48  
47  
46  
49  
47  
48  
46  
47  
48  
48  
48  
48  
48  
50  
48  
47  
49  
46  
47  
47  
48



*Figure B13.* NFS Large File Copy (1 of 2)  
oscarnode1.home.net

50  
52  
51  
54  
53  
50  
50  
51  
51  
57  
52  
51  
51  
51  
50  
50  
53  
51  
51  
51  
55  
52  
52  
52  
50  
51  
52  
50  
51  
52  
51  
51  
51  
52  
52

*Figure B14.* NFS Large File Copy (2 of 2)  
oscarnode2.home.net

51  
50  
51  
54  
53  
50  
51  
50  
50  
56  
52  
51  
51  
51  
50  
50  
52  
51  
51  
49  
53  
52  
52  
52  
50  
51  
52  
50  
51  
52  
51  
51  
50  
51  
52

*Figure B15.* NFS Large File Copy (1 of 3)  
oscarnode1.home.net

69  
68  
68  
68  
68  
72  
68  
69  
66  
68  
68  
68  
68  
69  
68  
68  
69  
69  
67  
68  
68  
69  
69  
68  
67  
70  
71  
69  
67  
69  
67  
68  
68  
68  
69

*Figure B16.* NFS Large File Copy (2 of 3)  
oscarnode2.home.net

69  
69  
66  
68  
67  
72  
68  
67  
70  
68  
67  
69  
68  
69  
68  
68  
67  
68  
68  
67  
69  
69  
69  
68  
68  
69  
71  
69  
68  
68  
69  
68  
67  
70  
68

*Figure B17.* NFS Large File Copy (3 of 3)  
oscarnode3.home.net

69  
69  
66  
68  
67  
71  
67  
68  
69  
68  
67  
68  
67  
69  
69  
68  
69  
68  
68  
68  
69  
69  
70  
67  
68  
69  
71  
69  
68  
69  
68  
68  
69  
67  
70

*Figure B18.* NFS Large File Copy (1 of 4)  
oscarnode1.home.net

91  
86  
86  
86  
97  
101  
94  
97  
98  
101  
92  
96  
92  
96  
95  
95  
98  
95  
94  
91  
89  
109  
108  
113  
103  
114  
103  
93  
88  
86  
93  
90  
86  
90  
68

*Figure B19.* NFS Large File Copy (2 of 4)  
oscarnode2.home.net

89  
90  
89  
86  
94  
100  
95  
98  
94  
102  
92  
95  
92  
94  
98  
93  
100  
96  
93  
86  
89  
93  
98  
97  
92  
94  
94  
90  
83  
90  
89  
91  
90  
90  
86

*Figure B20. NFS Large File Copy (3 of 4)*  
oscarnode3.home.net

87  
88  
94  
94  
102  
114  
106  
106  
105  
116  
108  
97  
94  
98  
95  
97  
98  
91  
85  
90  
92  
97  
97  
94  
93  
94  
94  
93  
87  
93  
83  
86  
88  
86  
69



*Figure B21.* NFS Large File Copy (4 of 4)  
oscarnode4.home.net

89  
88  
89  
89  
91  
102  
94  
99  
100  
97  
92  
103  
106  
111  
111  
110  
107  
110  
91  
86  
93  
98  
97  
93  
94  
94  
96  
86  
90  
86  
87  
91  
90  
87  
68

*Figure B22.* PVFS Large File Copy (1 of 1)  
oscarnode1.home.net

51  
49  
50  
49  
49  
49  
49  
49  
50  
50  
49  
49  
50  
49  
49  
50  
50  
49  
49  
50  
49  
49  
50  
50  
49  
50  
50  
49  
49  
49

*Figure B23. PVFS Large File Copy (1 of 2)*  
oscarnode1.home.net

76  
77  
77  
76  
76  
76  
77  
76  
77  
77  
76  
76  
77  
76  
76  
76  
77  
76  
77  
75  
76  
77  
76  
77  
77  
76  
76  
76  
76  
76  
76  
76  
77  
76  
76  
77

*Figure B24. PFVS Large File Copy (2 of 2)*  
oscarnode2.home.net

77  
76  
77  
77  
76  
76  
76  
76  
77  
77  
76  
76  
77  
76  
77  
76  
76  
77  
77  
76  
77  
77  
76  
76  
77  
77  
76  
76  
77  
76  
76  
74

*Figure B25. PVFS Large File Copy (1 of 3)*

oscarnode1.home.net

103

104

101

104

101

104

102

102

102

103

102

102

103

101

102

103

100

104

101

102

104

102

104

105

101

101

101

104

101

103

105

102

102

102

102

*Figure B26. PVFS Large File Copy (2 of 3)*  
oscarnode2.home.net

103  
104  
100  
103  
101  
104  
102  
102  
103  
102  
101  
101  
102  
102  
103  
102  
99  
104  
102  
102  
104  
102  
104  
105  
101  
100  
102  
104  
101  
104  
105  
102  
101  
102  
102

*Figure B27. PVFS Large File Copy (3 of 3)*  
oscarnode3.home.net

103  
104  
101  
103  
101  
104  
102  
102  
102  
103  
102  
102  
103  
102  
103  
102  
102  
100  
104  
101  
103  
104  
102  
104  
105  
101  
100  
101  
103  
102  
103  
106  
102  
102  
102  
102

*Figure B28. PVFS Large File Copy (1 of 4)*  
oscarnode1.home.net

119  
126  
125  
129  
126  
128  
132  
127  
127  
127  
124  
129  
130  
132  
127  
127  
124  
126  
126  
124  
129  
124  
136  
121  
120  
121  
122  
120  
124  
125  
126  
125  
129  
131  
124



*Figure B29. PVFS Large File Copy (2 of 4)*  
oscarnode2.home.net

119  
126  
125  
129  
126  
127  
131  
127  
127  
127  
124  
129  
130  
132  
127  
127  
124  
125  
125  
125  
130  
124  
136  
122  
120  
122  
122  
120  
125  
125  
127  
125  
129  
131  
122

*Figure B30.* PVFS Large File Copy (3 of 4)  
oscarnode3.home.net

120  
126  
125  
129  
125  
127  
132  
127  
127  
126  
124  
129  
130  
132  
127  
127  
125  
126  
125  
125  
130  
124  
136  
121  
120  
121  
122  
120  
125  
125  
127  
124  
128  
131  
123

*Figure B31.* PVFS Large File Copy (4 of 4)  
oscarnode4.home.net

119  
126  
124  
129  
126  
128  
132  
127  
127  
126  
124  
129  
130  
132  
127  
127  
124  
126  
126  
125  
129  
124  
136  
122  
120  
122  
122  
121  
125  
125  
126  
125  
129  
131  
125

*Figure B32.* Lustre Small File Copy (1 of 1)

oscarnode1.home.net

27  
22  
23  
23  
22  
22  
23  
22  
22  
22  
22  
22  
23  
22  
22  
22  
22  
22  
23  
22  
22  
22  
22  
23  
22  
22  
22  
22  
22  
23  
22  
22  
22  
22  
22  
23

*Figure B33.* Lustre Small File Copy (1 of 2)

oscarnode1.home.net

22  
23  
23  
23  
23  
23  
22  
23  
23  
23  
23  
23  
23  
23  
23  
23  
22  
23  
23  
23  
23  
23  
23  
23  
23  
22  
23  
23  
23  
23  
23  
22  
22  
23

*Figure B34.* Lustre Small File Copy (2 of 2)  
oscarnode2.home.net

26  
23  
23  
23  
23  
22  
23  
23  
23  
23  
23  
22  
23  
23  
23  
23  
22  
23  
23  
23  
22  
23  
23  
23  
23  
22  
23  
23  
23  
23  
22  
23  
23  
23

*Figure B35. Lustre Small File Copy (1 of 3)*  
oscarnode1.home.net

28  
23  
24  
23  
24  
24  
24  
23  
24  
24  
24  
42  
44  
44  
44  
45  
59  
46  
44  
45  
44  
46  
63  
44  
45  
45  
46  
45  
63  
46  
45  
46  
44  
61  
48

*Figure B36.* Lustre Small File Copy (2 of 3)  
oscarnode2.home.net

28  
23  
24  
23  
24  
24  
24  
23  
24  
23  
25  
46  
45  
44  
45  
60  
45  
45  
44  
45  
46  
60  
46  
45  
45  
45  
61  
47  
46  
45  
46  
45  
62  
46  
46



*Figure B37. Lustre Small File Copy (3 of 3)*  
oscarnode3.home.net

32  
24  
24  
23  
24  
24  
24  
23  
24  
24  
23  
45  
44  
45  
59  
45  
45  
45  
45  
45  
45  
59  
47  
45  
45  
45  
45  
62  
45  
45  
46  
45  
45  
63  
45  
46

*Figure B38.* Lustre Small File Copy (1 of 4)  
oscarnode1.home.net

31  
115  
121  
115  
115  
116  
113  
101  
87  
79  
75  
75  
64  
68  
72  
55  
72  
54  
72  
54  
74  
70  
56  
70  
55  
73  
53  
73  
53  
73  
52  
74  
53  
73  
55

*Figure B39. Lustre Small File Copy (2 of 4)*  
oscarnode2.home.net

31  
113  
120  
113  
115  
113  
113  
105  
90  
77  
75  
75  
59  
73  
52  
75  
70  
55  
71  
54  
71  
54  
72  
53  
73  
52  
74  
51  
74  
52  
73  
70  
56  
70  
55

*Figure B40.* Lustre Small File Copy (3 of 4)  
oscarnode3.home.net

31  
113  
118  
116  
113  
114  
112  
103  
91  
79  
75  
75  
56  
72  
54  
73  
70  
57  
70  
55  
72  
54  
73  
53  
73  
52  
73  
52  
74  
52  
73  
67  
59  
71  
54

*Figure B41.* Lustre Small File Copy (4 of 4)  
oscarnode4.home.net

33  
120  
117  
114  
116  
114  
113  
103  
86  
77  
76  
74  
56  
73  
53  
74  
70  
55  
70  
55  
73  
53  
73  
53  
73  
68  
58  
71  
55  
71  
54  
71  
54  
72  
54

*Figure B42.* NFS Small File Copy (1 of 1)

oscarnode1.home.net

22

21

21

19

19

20

21

19

21

20

19

20

20

21

19

21

20

20

20

20

20

20

20

19

20

22

20

20

20

20

20

21

19

20

21

*Figure B43.* NFS Small File Copy (1 of 2)  
oscarnode1.home.net

27  
29  
30  
32  
34  
36  
35  
35  
35  
36  
35  
36  
36  
36  
40  
35  
37  
35  
36  
35  
37  
37  
40  
37  
39  
39  
39  
40  
39  
41  
41  
39  
41  
41  
41  
41

*Figure B44.* NFS Small File Copy (2 of 2)  
oscarnode2.home.net

28  
29  
31  
34  
36  
35  
35  
35  
36  
36  
35  
37  
36  
38  
36  
36  
36  
35  
37  
35  
37  
38  
40  
38  
40  
38  
40  
40  
40  
40  
40  
41  
40  
42  
41  
40  
37



*Figure B45.* NFS Small File Copy (1 of 3)  
oscarnode1.home.net

53  
50  
53  
53  
55  
58  
56  
57  
55  
54  
55  
55  
53  
53  
54  
53  
54  
54  
53  
54  
53  
55  
56  
55  
54  
53  
54  
54  
59  
54  
54  
53  
55  
54  
55

*Figure B46.* NFS Small File Copy (2 of 3)  
oscarnode2.home.net

54  
50  
53  
54  
55  
58  
57  
57  
56  
55  
54  
56  
54  
53  
54  
53  
53  
56  
54  
54  
54  
53  
55  
57  
55  
54  
55  
54  
59  
55  
55  
54  
54  
55  
55  
51

*Figure B47.* NFS Small File Copy (3 of 3)  
oscarnode3.home.net

55  
53  
55  
58  
58  
61  
58  
56  
55  
55  
54  
56  
52  
55  
52  
53  
55  
54  
52  
54  
53  
54  
56  
55  
53  
55  
52  
60  
54  
55  
53  
55  
55  
56  
46

*Figure B48.* NFS Small File Copy (1 of 4)  
oscarnode1.home.net

69  
68  
68  
70  
68  
72  
69  
70  
71  
71  
70  
69  
70  
71  
70  
70  
70  
69  
69  
70  
69  
69  
69  
68  
69  
71  
70  
68  
69  
70  
69  
71  
70  
70  
70

*Figure B49.* NFS Small File Copy (2 of 4)  
oscarnode2.home.net

70  
67  
69  
70  
68  
72  
70  
69  
72  
70  
71  
69  
71  
71  
70  
70  
70  
72  
68  
69  
69  
70  
68  
69  
70  
71  
69  
69  
69  
70  
72  
69  
69  
71  
69

*Figure B50.* NFS Small File Copy (3 of 4)  
oscarnode3.home.net

70  
69  
68  
69  
69  
72  
70  
70  
71  
70  
71  
69  
70  
71  
70  
71  
69  
72  
68  
69  
69  
70  
69  
69  
69  
71  
70  
68  
70  
70  
71  
69  
69  
71  
69

*Figure B51.* NFS Small File Copy (4 of 4)  
oscarnode4.home.net

71  
69  
68  
71  
70  
73  
70  
72  
72  
71  
70  
71  
71  
71  
71  
71  
70  
72  
69  
70  
70  
70  
69  
70  
71  
70  
70  
69  
70  
71  
72  
70  
70  
73  
52

*Figure B52.* PVFS Small File Copy (1 of 1)  
oscarnode1.home.net

54  
54  
54  
55  
54  
56  
54  
55  
54  
55  
54  
55  
54  
55  
54  
55  
54  
55  
55  
54  
55  
54  
54  
55  
55  
54  
55  
55  
54  
55  
54  
55  
54  
55  
55



*Figure B53.* PVFS Small File Copy (1 of 2)  
oscarnode1.home.net

69  
70  
71  
73  
73  
73  
72  
74  
74  
74  
74  
73  
73  
74  
74  
74  
72  
73  
73  
74  
74  
75  
74  
73  
75  
74  
74  
74  
74  
74  
74  
74  
74  
74  
71

*Figure B54.* PVFS Small File Copy (2 of 2)  
oscarnode2.home.net

65  
69  
71  
72  
74  
72  
73  
74  
74  
73  
74  
72  
74  
74  
74  
73  
72  
74  
73  
73  
74  
74  
74  
73  
75  
73  
75  
73  
74  
74  
73  
74  
74  
74  
73

*Figure B55.* PVFS Small File Copy (1 of 3)

oscarnode1.home.net

84

84

89

89

90

90

90

89

89

89

89

90

88

89

89

89

90

89

89

89

89

89

91

88

89

89

89

90

87

89

89

89

89

90

79

*Figure B56.* PVFS Small File Copy (2 of 3)  
oscarnode2.home.net

82  
84  
88  
89  
89  
89  
88  
88  
88  
88  
88  
88  
87  
88  
88  
88  
88  
89  
88  
88  
88  
89  
88  
88  
88  
88  
87  
88  
89  
89  
88  
88  
88  
88  
89  
89  
88

*Figure B57.* PVFS Small File Copy (3 of 3)  
oscarnode3.home.net

76  
85  
88  
89  
89  
90  
89  
88  
89  
89  
89  
89  
88  
89  
89  
88  
89  
89  
89  
88  
89  
89  
87  
89  
86  
89  
90  
87  
88  
88  
88  
88  
89  
89  
89

*Figure B58.* PVFS Small File Copy (1 of 4)

oscarnode1.home.net

102

109

112

110

110

109

109

109

110

110

109

110

110

110

110

110

111

111

110

111

111

110

113

111

113

113

115

114

113

113

112

112

110

110

75

*Figure B59.* PVFS Small File Copy (2 of 4)  
oscarnode2.home.net

100  
106  
107  
109  
109  
108  
109  
107  
109  
108  
108  
108  
107  
108  
111  
109  
108  
109  
109  
107  
110  
109  
110  
110  
110  
111  
111  
112  
112  
111  
110  
110  
109  
108  
101

*Figure B60.* PVFS Small File Copy (3 of 4)  
oscarnode3.home.net

101  
106  
110  
110  
110  
108  
109  
109  
107  
109  
107  
109  
109  
107  
111  
109  
109  
109  
109  
108  
110  
110  
110  
110  
111  
111  
113  
111  
112  
113  
111  
111  
110  
110  
95



*Figure B61.* PVFS Small File Copy (4 of 4)  
oscarnode4.home.net

89  
107  
107  
107  
108  
107  
108  
106  
106  
107  
107  
106  
106  
108  
107  
108  
108  
106  
108  
108  
108  
107  
108  
106  
108  
108  
109  
110  
109  
110  
111  
110  
109  
109  
109  
108  
107

*Figure B62.* Lustre Small File Write (1 of 1)  
oscarnode1.home.net

15  
10  
10  
13  
10  
10  
12  
10  
10  
10  
11  
11  
10  
10  
11  
10  
11  
10  
10  
10  
10  
10  
11  
11  
11  
11  
11  
11  
10  
10  
12  
10  
11  
11  
11  
10

*Figure B63.* Lustre Small File Write (1 of 2)  
oscarnode1.home.net

11  
10  
10  
13  
11  
10  
12  
11  
10  
11  
12  
11  
10  
11  
12  
11  
11  
11  
12  
11  
11  
11  
12  
11  
11  
11  
11  
11  
11  
11  
11  
11  
11  
11  
12  
11

*Figure B64.* Lustre Small File Write (2 of 2)  
oscarnode2.home.net

11  
10  
10  
14  
10  
11  
11  
13  
10  
10  
13  
11  
11  
11  
12  
11  
10  
11  
11  
11  
11  
12  
12  
11  
11  
12  
11  
11  
12  
11  
11  
12  
11  
11  
12  
11  
11

*Figure B65.* Lustre Small File Write (1 of 3)  
oscarnode1.home.net

11  
10  
11  
14  
10  
11  
11  
12  
10  
11  
12  
11  
11  
11  
12  
10  
11  
11  
12  
11  
11  
12  
11  
10  
11  
12  
11  
11  
11  
11  
11  
11  
12  
11  
11

11  
10  
10  
13  
10  
10  
12  
11  
10  
11  
12  
11  
10  
11  
12  
11  
11  
12  
12  
11  
11  
12  
11  
11  
12  
11  
11  
12  
11  
11  
11  
11  
11  
11  
11

*Figure B67.* Lustre Small File Write (3 of 3)  
oscarnode3.home.net

10  
10  
10  
13  
11  
10  
12  
12  
10  
11  
12  
11  
11  
12  
11  
11  
11  
11  
12  
11  
11  
11  
11  
11  
11  
11  
11  
11  
12  
11  
11  
12  
11  
11  
11  
12  
11

*Figure B68.* Lustre Small File Write (1 of 4)  
oscarnode1.home.net

11  
11  
10  
16  
10  
10  
14  
10  
11  
11  
11  
10  
12  
11  
11  
12  
11  
11  
12  
11  
10  
12  
11  
11  
13  
11  
11  
11  
12  
10  
11  
12  
11  
10  
12



*Figure B69.* Lustre Small File Write (2 of 4)  
oscarnode2.home.net

11  
11  
10  
15  
10  
11  
13  
10  
11  
11  
12  
11  
11  
12  
11  
11  
11  
12  
11  
12  
11  
12  
11  
10  
11  
11  
11  
12  
11  
12  
11  
12  
11  
12  
11  
12  
11  
11  
11  
11

*Figure B70.* Lustre Small File Copy (3 of 4)  
oscarnode3.home.net

10  
11  
10  
16  
10  
11  
14  
11  
11  
11  
12  
10  
12  
12  
11  
11  
12  
11  
12  
11  
11  
12  
12  
11  
12  
11  
11  
11  
11  
12  
11  
11  
12  
11  
11  
11  
11

*Figure B71.* Lustre Small File Write (4 of 4)  
oscarnode4.home.net

11

10

15

10

10

12

10

10

10

10

10

10

10

10

10

10

10

10

10

10

10

10

10

10

10

10

*Figure B72.* NFS Small File Write (1 of 1)

oscarnode1.home.net

12  
10  
10  
10  
11  
11  
10  
11  
11  
11  
11  
10  
10  
11  
11  
10  
10  
10  
10  
11  
13  
11  
10  
10  
10  
10  
10  
10  
11  
11  
10  
11  
10  
10  
10  
11  
10  
11

*Figure B73.* NFS Small File Write (1 of 2)  
oscarnode1.home.net

19  
20  
20  
21  
20  
21  
21  
19  
20  
20  
21  
20  
19  
21  
20  
20  
21  
20  
20  
21  
20  
20  
20  
20  
20  
20  
20  
19  
20  
20  
20  
20  
19  
20  
19  
19  
20

*Figure B74.* NFS Small File Write (2 of 2)  
oscarnode2.home.net

20  
20  
20  
21  
20  
21  
21  
19  
20  
21  
20  
20  
19  
20  
20  
21  
21  
20  
22  
21  
21  
21  
20  
20  
20  
20  
19  
20  
20  
21  
20  
20  
20  
20  
20

*Figure B75.* NFS Small File Write (1 of 3)  
oscarnode1.home.net

29  
29  
29  
29  
29  
29  
30  
29  
30  
29  
30  
29  
30  
29  
29  
30  
29  
29  
29  
31  
29  
28  
28  
28  
29  
29  
29  
29  
29  
29  
29  
29  
29  
29  
29

*Figure B76.* NFS Small File Write (2 of 3)  
oscarnode2.home.net

29  
29  
29  
29  
29  
29  
30  
30  
30  
29  
30  
30  
29  
29  
30  
29  
28  
29  
31  
30  
29  
30  
29  
28  
29  
30  
29  
30  
30  
29  
29  
30  
29  
30  
29



*Figure B77.* NFS Small File Write (3 of 3)  
oscarnode3.home.net

29  
29  
29  
29  
29  
28  
29  
29  
29  
29  
29  
30  
29  
29  
30  
28  
29  
29  
31  
29  
28  
29  
28  
29  
30  
29  
29  
29  
29  
30  
29  
29  
29

*Figure B78.* NFS Small File Write (1 of 4)

oscarnode1.home.net

35

37

36

36

36

37

37

35

36

35

35

36

36

37

36

35

35

36

36

37

35

39

37

36

35

36

36

36

37

37

40

37

36

36

36

*Figure B79.* NFS Small File Write (2 of 4)  
oscarnode2.home.net

35  
37  
36  
36  
37  
37  
37  
36  
37  
36  
36  
36  
37  
36  
37  
35  
36  
36  
36  
35  
36  
39  
37  
37  
35  
36  
36  
36  
36  
36  
36  
40  
37  
36  
36  
36

*Figure B80.* NFS Small File Write (3 of 4)  
oscarnode3.home.net

35  
35  
36  
36  
37  
36  
37  
36  
36  
35  
37  
36  
36  
35  
36  
37  
35  
35  
36  
36  
35  
39  
37  
36  
36  
36  
36  
36  
36  
36  
37  
40  
37  
36  
36  
36

*Figure B81.* NFS Small File Write (4 of 4)  
oscarnode4.home.net

35  
36  
36  
36  
36  
37  
36  
35  
36  
36  
35  
36  
36  
36  
36  
37  
36  
36  
35  
36  
36  
36  
36  
39  
37  
36  
35  
36  
36  
36  
37  
36  
36  
40  
37  
36  
36  
36

28

27

*Figure B83. PVFS Small File Write (1 of 2)*

oscarnode1.home.net

31  
32  
31  
31  
32  
32  
32  
32  
33  
32  
31  
32  
31  
31  
32  
31  
31  
32  
32  
31  
31  
31  
31  
31  
31  
30  
30  
30  
30  
29  
29  
29  
28  
29  
29  
29  
29  
29  
29  
29  
29

*Figure B84.* PVFS Small File Write (2 of 2)  
oscarnode2.home.net

30  
31  
31  
32  
32  
31  
32  
32  
31  
32  
31  
31  
31  
31  
32  
32  
32  
30  
30  
31  
31  
30  
30  
30  
30  
29  
29  
29  
29  
28  
29  
29  
29  
28  
29  
28



*Figure B85. PVFS Small File Write (1 of 3)*

oscarnode1.home.net

39  
39  
38  
37  
38  
39  
38  
39  
39  
38  
39  
39  
39  
37  
38  
39  
39  
38  
39  
39  
37  
38  
39  
37  
39  
38  
39  
38  
39  
39  
39  
38  
38

*Figure B86.* PVFS Small File Write (2 of 3)  
oscarnode2.home.net

40  
39  
38  
37  
39  
39  
38  
38  
39  
38  
38  
39  
38  
39  
38  
38  
39  
38  
38  
38  
38  
39  
39  
37  
38  
38  
38  
39  
39  
39  
39  
38  
39

*Figure B87.* PVFS Small File Write (3 of 3)  
oscarnode3.home.net

39  
39  
38  
37  
38  
39  
38  
39  
38  
38  
39  
39  
38  
38  
39  
39  
39  
38  
39  
38  
37  
39  
38  
37  
39  
39  
37  
39  
39  
38  
39  
39  
38  
37  
39  
39

*Figure B88.* PVFS Small File Write (1 of 4)  
oscarnode1.home.net

38  
40  
40  
41  
42  
41  
40  
41  
42  
41  
40  
40  
41  
41  
41  
40  
40  
42  
42  
40  
41  
41  
41  
41  
41  
42  
42  
42  
41  
41  
42  
41  
42  
43  
41

*Figure B89.* PVFS Small File Write (2 of 4)  
oscarnode2.home.net

37  
40  
40  
41  
42  
41  
41  
41  
40  
40  
41  
41  
40  
41  
40  
40  
41  
41  
41  
40  
41  
41  
41  
41  
41  
42  
41  
42  
41  
41  
41  
41  
42  
42  
41

*Figure B90.* PVFS Small File Write (3 of 4)  
oscarnode3.home.net

36  
39  
41  
40  
41  
41  
41  
41  
40  
40  
40  
41  
40  
40  
40  
40  
40  
42  
41  
40  
41  
40  
42  
41  
41  
41  
41  
43  
41  
41  
42  
42  
40  
42  
41  
42

*Figure B91.* PVFS Small File Write (4 of 4)  
oscarnode4.home.net

35  
39  
40  
40  
41  
40  
41  
41  
40  
41  
41  
40  
40  
41  
39  
40  
41  
41  
40  
40  
41  
41  
40  
41  
40  
40  
41  
42  
41  
41  
41  
42  
40  
41  
41

## Appendix C

### File-System Benchmarks for virtual cluster nodes

*Figure C1.* Lustre Large File Copy (1 of 1)

oscarnode5.home.net

54  
60  
60  
57  
54  
54  
69  
79  
61  
60  
64  
60  
56  
61  
57  
57  
61  
60  
59  
64  
60  
60  
57  
65  
58  
63  
56  
58  
62  
59  
60  
63  
63  
55  
58



*Figure C2. Lustre Large File Copy (1 of 2)*  
oscarnode5.home.net

73  
70  
71  
71  
71  
68  
69  
70  
68  
70  
70  
70  
76  
73  
71  
72  
69  
79  
68  
69  
72  
71  
71  
67  
70  
72  
67  
72  
71  
73  
68  
68  
71  
73  
71

*Figure C3. Lustre Large File Copy (2 of 2)*  
oscarnode6.home.net

73  
71  
71  
70  
76  
70  
68  
73  
71  
72  
70  
70  
77  
74  
71  
68  
70  
79  
73  
69  
73  
72  
70  
73  
69  
66  
71  
71  
73  
70  
71  
70  
73  
68  
67

*Figure C4. Lustre Large File Copy (1 of 3)*  
oscarnode5.home.net

118  
109  
108  
101  
103  
99  
94  
119  
117  
99  
93  
92  
97  
98  
101  
96  
92  
92  
95  
99  
99  
98  
100  
96  
104  
100  
94  
99  
101  
101  
101  
108  
101  
99  
69

*Figure C5. Lustre Large File Copy (2 of 3)*  
oscarnode6.home.net

123  
108  
104  
103  
92  
95  
107  
121  
117  
100  
108  
99  
100  
103  
94  
101  
103  
107  
111  
100  
101  
103  
102  
96  
99  
105  
102  
111  
102  
106  
93  
95  
97  
78  
59

*Figure C6. Lustre Large File Copy (3 of 3)*  
oscarnode7.home.net

87  
89  
89  
97  
95  
102  
104  
94  
82  
92  
92  
101  
107  
103  
99  
99  
100  
104  
99  
100  
106  
99  
99  
100  
100  
108  
94  
104  
97  
94  
94  
94  
103  
97  
97

*Figure C7. Lustre Large File Copy (1 of 4)*  
oscarnode5.home.net

104  
105  
107  
105  
101  
103  
106  
107  
108  
123  
116  
106  
104  
101  
116  
101  
115  
108  
116  
115  
107  
106  
103  
99  
104  
110  
108  
102  
99  
100  
99  
118  
109  
107  
103

*Figure C8. Lustre Large File Copy (2 of 4)*  
oscarnode6.home.net

104  
104  
106  
106  
109  
109  
107  
109  
107  
139  
108  
106  
109  
110  
103  
118  
102  
102  
108  
106  
102  
107  
102  
105  
117  
102  
98  
116  
107  
109  
105  
115  
109  
101  
103

*Figure C9. Lustre Large File Copy (3 of 4)*  
oscarnode7.home.net

105  
101  
115  
110  
100  
103  
111  
104  
105  
136  
104  
106  
107  
106  
107  
105  
104  
100  
103  
104  
109  
103  
109  
114  
106  
106  
107  
117  
106  
110  
109  
116  
112  
109  
105



*Figure C10.* Lustre Large File Copy (4 of 4)  
oscarnode8.home.net

106  
113  
124  
106  
113  
113  
121  
113  
136  
124  
113  
113  
111  
123  
110  
116  
113  
124  
111  
108  
114  
120  
119  
114  
111  
132  
107  
107  
112  
116  
119  
116  
94  
61  
59

*Figure C11.* NFS Large File Copy (1 of 1)  
oscarnode5.home.net

79  
77  
74  
75  
78  
77  
82  
77  
75  
77  
77  
76  
76  
79  
76  
76  
76  
76  
76  
78  
77  
76  
77  
78  
75  
77  
76  
76  
76  
76  
78  
76  
79  
77  
77  
75  
76

*Figure C12.* NFS Large File Copy (1 of 2)  
oscarnode5.home.net

136  
142  
106  
106  
116  
110  
115  
107  
111  
103  
106  
103  
105  
103  
101  
102  
103  
107  
105  
103  
101  
108  
106  
115  
114  
110  
118  
111  
110  
105  
105  
111  
110  
111  
115

*Figure C13.* NFS Large File Copy (2 of 2)  
oscarnode6.home.net

137  
143  
107  
107  
116  
111  
116  
108  
112  
104  
106  
104  
105  
103  
101  
101  
104  
107  
105  
102  
101  
108  
106  
114  
113  
110  
118  
112  
109  
105  
105  
110  
109  
110  
114

*Figure C14.* NFS Large File Copy (1 of 3)  
oscarnode5.home.net

168  
166  
166  
181  
180  
167  
171  
174  
168  
166  
170  
166  
168  
172  
172  
181  
192  
180  
178  
188  
180  
166  
178  
187  
188  
175  
202  
172  
173  
185  
172  
165  
191  
197  
157

*Figure C15. NFS Large File Copy (2 of 3)*  
oscarnode6.home.net

169  
168  
173  
180  
169  
169  
175  
168  
168  
166  
171  
163  
171  
173  
171  
162  
164  
182  
182  
169  
168  
184  
181  
170  
173  
184  
164  
166  
185  
169  
185  
190  
199  
194  
190

*Figure C16.* NFS Large File Copy (3 of 3)  
oscarnode7.home.net

189  
179  
184  
180  
174  
184  
190  
195  
197  
192  
191  
175  
192  
170  
181  
193  
191  
194  
185  
199  
191  
199  
182  
182  
192  
200  
193  
181  
191  
184  
156  
175  
176  
100  
90

*Figure C17.* NFS Large File Copy (1 of 4)  
oscarnode5.home.net

184  
173  
167  
166  
163  
182  
179  
170  
164  
166  
168  
165  
163  
175  
184  
183  
159  
157  
158  
166  
190  
171  
171  
160  
165  
163  
174  
174  
170  
167  
198  
194  
167  
171  
168



*Figure C18.* NFS Large File Copy (2 of 4)  
oscarnode6.home.net

192  
185  
161  
163  
193  
160  
160  
172  
158  
171  
160  
177  
164  
178  
182  
162  
174  
163  
158  
184  
192  
167  
155  
165  
167  
172  
171  
179  
158  
197  
186  
176  
164  
168  
168

*Figure C19.* NFS Large File Copy (3 of 4)  
oscarnode7.home.net

185  
167  
163  
171  
170  
183  
180  
164  
166  
168  
161  
173  
171  
185  
168  
178  
163  
167  
163  
191  
185  
169  
154  
155  
170  
159  
163  
186  
168  
170  
222  
169  
168  
171  
166

*Figure C20. NFS Large File Copy (4 of 4)*  
oscarnode8.home.net

188  
173  
169  
161  
193  
174  
166  
162  
161  
174  
175  
157  
171  
169  
188  
174  
168  
169  
166  
179  
178  
161  
166  
168  
184  
164  
176  
173  
157  
187  
193  
184  
161  
157  
167

*Figure C21. PVFS Large File Copy (1 of 1)*  
oscarnode5.home.net

59  
56  
56  
57  
56  
56  
55  
53  
54  
67  
55  
54  
55  
55  
58  
55  
56  
57  
57  
61  
58  
58  
65  
57  
57  
60  
62  
62  
60  
64  
64  
59  
60  
62  
57

*Figure C22. PVFS Large File Copy (1 of 2)*  
oscarnode5.home.net

85  
87  
80  
88  
83  
79  
80  
73  
75  
71  
73  
76  
80  
76  
78  
77  
75  
74  
74  
75  
77  
75  
80  
76  
77  
76  
70  
75  
73  
78  
82  
76  
72  
72  
71

*Figure C23. PVFS Large File Copy (1 of 3)*  
oscarnode5.home.net

117  
107  
104  
111  
99  
97  
93  
93  
100  
93  
94  
97  
97  
100  
99  
97  
98  
98  
99  
98  
94  
94  
95  
92  
93  
95  
96  
102  
100  
99  
95  
100  
99  
100  
99

*Figure C24. PVFS Large File Copy (2 of 3)*  
oscarnode6.home.net

130  
134  
109  
103  
103  
109  
113  
111  
110  
110  
109  
104  
109  
101  
113  
111  
112  
110  
117  
114  
112  
110  
109  
109  
108  
106  
101  
105  
106  
103  
101  
86  
73  
68  
52

*Figure C25. PVFS Large File Copy (3 of 3)*  
oscarnode7.home.net

117  
117  
116  
122  
116  
108  
107  
105  
111  
111  
115  
113  
106  
110  
107  
100  
101  
105  
109  
109  
109  
99  
96  
100  
93  
97  
99  
101  
101  
94  
97  
97  
85  
68  
71



*Figure C26. PVFS Large File Copy (1 of 4)*  
oscarnode5.home.net

103  
100  
104  
96  
94  
102  
103  
102  
105  
101  
105  
101  
106  
100  
100  
105  
101  
101  
103  
99  
103  
98  
103  
100  
104  
105  
106  
98  
95  
95  
95  
97  
98  
101  
105

*Figure C27. PVFS Large File Copy (2 of 4)*  
oscarnode6.home.net

102  
101  
103  
111  
111  
107  
111  
106  
108  
108  
105  
104  
104  
118  
114  
105  
106  
111  
114  
110  
107  
102  
99  
110  
107  
114  
109  
109  
110  
110  
108  
101  
100  
70  
52

*Figure C28. PVFS Large File Copy (3 of 4)*  
oscarnode7.home.net

102  
94  
93  
96  
105  
104  
99  
98  
98  
100  
96  
110  
108  
104  
112  
102  
103  
99  
102  
97  
98  
97  
101  
108  
102  
100  
109  
98  
94  
92  
101  
99  
102  
101  
105

*Figure C29. PVFS Large File Copy (4 of 4)*  
oscarnode8.home.net

128  
124  
120  
107  
104  
96  
96  
102  
102  
96  
99  
113  
109  
113  
99  
98  
105  
114  
107  
114  
102  
103  
108  
99  
108  
99  
95  
103  
98  
97  
97  
97  
97  
98  
67

*Figure C30. Lustre Small File Copy (1 of 1)*  
oscarnode5.home.net

27  
25  
26  
24  
24  
26  
23  
24  
24  
25  
25  
24  
25  
25  
24  
25  
25  
24  
24  
24  
23  
25  
24  
25  
25  
24  
23  
24  
24  
24  
24  
24  
23  
24  
25  
24  
25  
24

*Figure C31. Lustre Small File Copy (1 of 2)*  
oscarnode5.home.net

28  
29  
26  
28  
27  
29  
27  
29  
27  
27  
28  
29  
27  
28  
27  
27  
27  
28  
30  
26  
27  
26  
27  
26  
28  
26  
27  
27  
29  
28  
29  
28  
29  
26  
27  
27

*Figure C34. Lustre Small File Copy (2 of 2)*  
oscarnode6.home.net

31  
33  
28  
28  
29  
28  
29  
28  
29  
27  
29  
28  
29  
28  
28  
28  
29  
30  
27  
28  
28  
28  
27  
29  
27  
28  
28  
30  
31  
30  
28  
29  
28  
28  
26  
26

*Figure C35. Lustre Small File Copy (1 of 3)*  
oscarnode5.home.net

44  
34  
44  
44  
43  
46  
48  
46  
46  
47  
45  
44  
46  
47  
49  
46  
48  
47  
48  
47  
48  
48  
48  
47  
48  
48  
47  
49  
47  
48  
48  
47  
48  
48  
46



*Figure C36. Lustre Small File Copy (2 of 3)*  
oscarnode6.home.net

45  
33  
48  
48  
48  
48  
47  
48  
46  
44  
46  
47  
45  
47  
46  
48  
48  
48  
47  
48  
48  
48  
48  
48  
47  
49  
47  
48  
48  
48  
48  
47  
48  
45

*Figure C37. Lustre Small File Copy (3 of 3)*  
oscarnode7.home.net

39  
33  
45  
47  
46  
44  
42  
43  
42  
47  
47  
45  
45  
47  
48  
48  
48  
48  
47  
48  
47  
48  
47  
48  
48  
48  
48  
48  
48  
48  
47  
48  
47  
48  
47  
48

*Figure C38. Lustre Small File Copy (1 of 4)*  
oscarnode5.home.net

49  
123  
119  
108  
98  
98  
97  
97  
93  
82  
77  
74  
73  
68  
68  
66  
66  
65  
65  
64  
65  
65  
65  
65  
64  
65  
65  
65  
64  
65  
65  
62  
66  
65  
64  
65

*Figure C39. Lustre Small File Copy (2 of 4)*  
oscarnode6.home.net

50  
146  
120  
106  
99  
97  
97  
97  
90  
84  
77  
74  
71  
70  
69  
68  
67  
65  
65  
67  
66  
66  
65  
66  
66  
66  
65  
65  
66  
67  
66  
65  
65  
66  
65  
63

*Figure C40.* Lustre Small File Copy (3 of 4)  
oscarnode7.home.net

75  
120  
120  
107  
96  
94  
96  
97  
89  
83  
78  
73  
73  
69  
68  
68  
65  
66  
66  
65  
65  
65  
67  
65  
65  
65  
66  
66  
65  
64  
66  
66  
65  
65  
64

*Figure C41.* Lustre Small File Copy (4 of 4)  
oscarnode8.home.net

49  
126  
120  
109  
98  
99  
99  
97  
92  
82  
80  
75  
72  
70  
70  
68  
67  
65  
66  
67  
65  
66  
66  
66  
65  
67  
66  
66  
65  
66  
65  
65  
67  
66  
66  
63

*Figure C42.* NFS Small File Copy (1 of 1)  
oscarnode5.home.net

31  
30  
30  
30  
30  
29  
30  
30  
29  
30  
29  
29  
30  
30  
29  
29  
29  
29  
29  
30  
29  
29  
29  
29  
30  
29  
29  
30  
28  
30  
29  
29  
29  
30  
30

*Figure C43. NFS Small File Copy (1 of 2)*  
oscarnode5.home.net

36  
36  
35  
35  
35  
36  
36  
36  
36  
36  
35  
35  
35  
36  
36  
36  
36  
35  
36  
35  
35  
36  
35  
35  
36  
36  
35  
36  
38  
36  
36  
35  
35  
36  
36  
35



*Figure C44.* NFS Small File Copy (2 of 2)  
oscarnode6.home.net

36  
36  
35  
36  
36  
36  
35  
36  
36  
36  
36  
35  
36  
36  
35  
36  
36  
36  
36  
36  
35  
36  
35  
37  
35  
36  
36  
36  
35  
40  
36  
36  
36  
36  
36  
37  
36  
34

*Figure C45. NFS Small File Copy (1 of 3)*  
oscarnode5.home.net

47  
47  
46  
48  
48  
49  
48  
46  
48  
47  
48  
49  
48  
48  
49  
49  
47  
48  
49  
49  
48  
48  
48  
48  
46  
47  
47  
46  
46  
48  
48  
47  
48  
37  
37

*Figure C46.* NFS Small File Copy (2 of 3)  
oscarnode6.home.net

48  
49  
49  
49  
49  
50  
48  
48  
48  
49  
49  
48  
49  
49  
50  
48  
49  
45  
45  
47  
49  
49  
49  
49  
47  
47  
47  
46  
48  
48  
48  
48  
45  
37  
35

*Figure C47. NFS Small File Copy (3 of 3)*  
oscarnode7.home.net

45  
47  
46  
46  
47  
45  
44  
43  
44  
46  
45  
44  
45  
44  
44  
45  
44  
44  
46  
48  
46  
44  
45  
43  
45  
44  
45  
47  
48  
45  
43  
44  
44  
44  
44

*Figure C48.* NFS Small File Copy (1 of 4)  
oscarnode5.home.net

57  
59  
58  
59  
58  
55  
56  
56  
56  
56  
56  
54  
59  
58  
58  
55  
55  
57  
57  
58  
58  
54  
54  
54  
54  
54  
54  
53  
53  
52  
53  
53  
54  
55  
54  
53

*Figure C49. NFS Small File Copy (2 of 4)*  
oscarnode6.home.net

58  
59  
59  
60  
58  
56  
56  
58  
57  
57  
56  
55  
51  
56  
60  
56  
55  
57  
58  
57  
56  
59  
59  
61  
58  
59  
55  
55  
58  
60  
59  
59  
59  
56  
33

*Figure C50. NFS Small File Copy (3 of 4)*  
oscarnode7.home.net

54  
54  
53  
51  
56  
56  
55  
55  
57  
54  
56  
53  
56  
58  
56  
59  
55  
55  
58  
58  
58  
58  
58  
58  
58  
56  
58  
57  
57  
58  
58  
57  
58  
58  
49

*Figure C51.* NFS Small File Copy (4 of 4)  
oscarnode8.home.net

54  
54  
55  
58  
55  
55  
57  
57  
56  
55  
55  
55  
57  
59  
57  
60  
54  
53  
51  
52  
54  
54  
54  
55  
55  
55  
55  
59  
59  
55  
56  
54  
55  
55  
56



*Figure C52. Lustre Small File Write (1 of 1)*  
oscarnode5.home.net

15  
10  
10  
14  
10  
10  
12  
12  
10  
11  
12  
10  
11  
11  
12  
11  
11  
11  
12  
12  
11  
11  
12  
11  
11  
11  
12  
11  
11  
11  
12  
11  
12

*Figure C53. Luster Small File Write (1 of 2)*  
oscarnode5.home.net

11  
11  
12  
13  
11  
11  
13  
11  
12  
12  
12  
12  
13  
12  
11  
13  
12  
12  
13  
12  
12  
13  
12  
12  
12  
11  
11  
12  
11  
12  
12  
12  
11  
12  
11

*Figure C54. Lustre Small File Write (2 of 2)*  
oscarnode6.home.net

11  
11  
11  
13  
12  
12  
12  
11  
11  
12  
11  
11  
12  
11  
12  
12  
13  
11  
12  
13  
12  
12  
12  
12  
12  
13  
11  
12  
12  
12  
11  
12  
11  
12  
12  
12

*Figure C55. Lustre Small File Write (1 of 3)*  
oscarnode5.home.net

13  
13  
14  
14  
14  
14  
15  
14  
14  
14  
13  
15  
14  
13  
14  
14  
14  
13  
14  
14  
13  
13  
14  
13  
14  
14  
13  
14  
14  
13  
14  
14  
13  
15  
13  
15  
14  
14

*Figure C56. Lustre Small File Write (2 of 3)*  
oscarnode6.home.net

14  
13  
14  
14  
13  
14  
14  
13  
13  
14  
13  
14  
13  
13  
14  
14  
14  
14  
14  
14  
13  
15  
14  
14  
14  
14  
13  
14  
14  
14  
14  
13  
14  
14  
14  
12

*Figure C57. Lustre Small File Write (3 of 3)*  
oscarnode7.home.net

13  
13  
14  
13  
13  
14  
13  
13  
14  
14  
13  
14  
14  
13  
15  
15  
14  
15  
14  
14  
15  
14  
13  
14  
13  
13  
13  
13  
13  
13  
13  
14  
12  
13  
13

*Figure C58. Lustre Small File Write (1 of 4)*  
oscarnode5.home.net

15  
16  
20  
16  
18  
16  
17  
17  
18  
17  
17  
18  
17  
16  
18  
16  
16  
16  
18  
17  
17  
17  
17  
17  
17  
17  
17  
17  
17  
17  
16  
18  
17  
17  
18  
15

*Figure C59. Lustre Small File Write (2 of 4)*  
oscarnode6.home.net

16  
15  
20  
15  
18  
16  
16  
17  
16  
16  
17  
17  
18  
17  
17  
17  
18  
17  
18  
17  
17  
17  
17  
17  
18  
17  
17  
16  
17  
17  
17  
18  
17  
17  
17



*Figure C60.* Lustre Small File Write (3 of 4)  
oscarnode7.home.net

17  
16  
20  
16  
18  
16  
18  
17  
18  
17  
17  
16  
16  
17  
16  
17  
17  
16  
17  
17  
16  
18  
16  
17  
16  
17  
17  
17  
17  
17  
17  
18  
17  
18  
17  
18  
17

*Figure C61.* Lustre Small File Write (4 of 4)  
oscarnode8.home.net

15  
16  
20  
16  
17  
17  
17  
18  
17  
17  
17  
18  
18  
17  
19  
18  
16  
18  
18  
18  
18  
18  
18  
18  
18  
17  
19  
18  
18  
18  
18  
17  
18  
18  
18  
18  
17

*Figure C62.* NFS Small File Write (1 of 1)  
oscarnode5.home.net

15  
14  
14  
13  
14  
14  
14  
13  
14  
14  
13  
13  
14  
14  
13  
13  
13  
14  
13  
13  
13  
14  
14  
13  
14  
13  
13  
14  
14  
13  
13  
14  
14

*Figure C63. NFS Small File Write (1 of 2)*  
oscarnode5.home.net

17  
17  
18  
17  
17  
17  
18  
17  
17  
17  
17  
17  
17  
17  
17  
17  
18  
18  
17  
17  
18  
18  
17  
17  
17  
18  
21  
18  
18  
18  
17  
18  
18  
17  
17

*Figure C64. NFS Small File Write (2 of 2)*  
oscarnode6.home.net

17  
17  
18  
17  
18  
18  
18  
18  
18  
18  
19  
18  
19  
18  
18  
18  
17  
17  
17  
17  
18  
18  
18  
18  
18  
18  
18  
21  
18  
18  
17  
17  
17  
17  
18  
18

*Figure C65. NFS Small File Write (1 of 3)*

oscarnode5.home.net

22  
23  
22  
23  
22  
24  
24  
24  
23  
23  
23  
24  
24  
24  
24  
24  
24  
25  
24  
23  
23  
24  
24  
24  
24  
24  
24  
25  
24  
25  
24  
23  
25  
24  
24  
24

*Figure C66. NFS Small File Write (2 of 3)*

oscarnode6.home.net

24  
25  
24  
24  
24  
24  
25  
25  
24  
25  
25  
24  
24  
25  
25  
25  
24  
24  
24  
25  
25  
24  
25  
25  
25  
24  
24  
25  
24  
24  
25  
25  
24  
24  
23

*Figure C67. NFS Small File Write (3 of 3)*  
oscarnode7.home.net

24  
24  
24  
24  
24  
21  
23  
22  
22  
22  
22  
22  
23  
21  
22  
22  
22  
22  
23  
22  
23  
23  
22  
22  
23  
22  
22  
22  
22  
22  
22  
22  
22  
21  
22  
22  
22



*Figure C68. NFS Small File Write (1 of 4)*  
oscarnode5.home.net

29  
29  
29  
29  
28  
29  
28  
29  
28  
29  
28  
28  
27  
31  
28  
28  
27  
27  
27  
27  
26  
26  
27  
27  
27  
29  
29  
29  
29  
28  
28  
29  
29

*Figure C69. NFS Small File Write (2 of 4)*  
oscarnode6.home.net

29  
27  
27  
28  
28  
28  
29  
30  
29  
29  
30  
28  
28  
28  
31  
28  
27  
28  
28  
28  
29  
28  
30  
30  
29  
29  
30  
29  
29  
29  
29  
29  
30  
29

*Figure C70. NFS Small File Write (3 of 4)*  
oscarnode7.home.net

28  
28  
28  
24  
24  
24  
25  
25  
26  
26  
26  
26  
28  
29  
28  
30  
27  
27  
27  
27  
27  
27  
26  
27  
27  
28  
28  
27  
27  
26  
26  
26  
27  
28  
26

*Figure C71.* NFS Small File Write (4 of 4)  
oscarnode8.home.net

26  
26  
27  
28  
29  
30  
30  
28  
26  
27  
27  
28  
29  
29  
31  
28  
28  
28  
28  
28  
27  
28  
29  
30  
29  
29  
28  
28  
28  
27  
27  
29  
29  
28  
28

*Figure C72. PVFS Small File Write (1 of 1)*  
oscarnode5.home.net

33  
29  
30  
29  
30  
29  
30  
29  
30  
29  
30  
28  
29  
29  
30  
29  
29  
29  
28  
28  
29  
30  
29  
29  
28  
29  
29  
30  
28  
29  
28  
29  
28  
29  
29  
30

*Figure C73. PVFS Small File Write (1 of 2)*  
oscarnode5.home.net

35  
38  
38  
40  
41  
40  
40  
39  
39  
39  
38  
39  
41  
39  
39  
39  
41  
40  
40  
41  
41  
40  
40  
40  
40  
40  
40  
39  
40  
41  
39  
41  
40  
39  
38  
43

*Figure C74. PVFS Small File Write (2 of 2)*  
oscarnode6.home.net

33  
38  
39  
41  
41  
40  
40  
41  
41  
39  
40  
41  
40  
40  
40  
41  
41  
40  
40  
41  
40  
41  
41  
40  
40  
41  
41  
42  
40  
40  
40  
41  
41  
40  
42

*Figure C75. PVFS Small File Write (1 of 3)*  
oscarnode5.home.net

53  
57  
54  
57  
55  
56  
57  
55  
56  
53  
56  
54  
55  
55  
57  
54  
52  
53  
54  
55  
53  
58  
56  
55  
56  
55  
57  
57  
54  
55  
57  
56  
55  
55  
55



*Figure C76. PVFS Small File Write (2 of 3)*  
oscarnode6.home.net

55  
57  
56  
57  
57  
57  
58  
55  
57  
55  
55  
55  
54  
52  
55  
57  
56  
57  
58  
60  
55  
57  
56  
56  
56  
57  
56  
58  
56  
58  
55  
58  
55  
57  
56

*Figure C77. PVFS Small File Write (3 of 3)*  
oscarnode7.home.net

51  
52  
53  
53  
53  
54  
52  
53  
52  
51  
51  
55  
55  
56  
55  
55  
55  
55  
56  
57  
56  
57  
54  
53  
55  
55  
56  
55  
53  
54  
51  
53  
53  
52  
52  
54

*Figure C78. PVFS Small File Write (1 of 4)*  
oscarnode5.home.net

60  
63  
64  
65  
65  
70  
69  
68  
65  
66  
66  
64  
64  
63  
70  
70  
70  
70  
70  
69  
68  
70  
68  
68  
68  
67  
69  
68  
67  
68  
69  
68  
67  
68  
67

*Figure C79.* PVFS Small File Write (2 of 4)  
oscarnode6.home.net

60  
65  
64  
67  
66  
61  
64  
69  
67  
69  
66  
70  
73  
71  
70  
72  
70  
70  
71  
70  
70  
72  
68  
69  
68  
69  
70  
69  
67  
63  
69  
68  
68  
69  
69

*Figure C80. PVFS Small File Copy (3 of 4)*  
oscarnode7.home.net

51  
63  
65  
66  
65  
69  
68  
67  
68  
71  
67  
71  
71  
73  
64  
68  
66  
66  
68  
68  
69  
70  
68  
68  
67  
67  
68  
67  
67  
67  
71  
67  
68  
68  
69

*Figure C81. PVFS Small File Write (4 of 4)*  
oscarnode8.home.net

55  
66  
65  
68  
68  
72  
68  
68  
69  
71  
70  
75  
75  
70  
68  
68  
67  
68  
69  
70  
71  
70  
71  
69  
71  
69  
70  
68  
67  
73  
68  
69  
67  
70  
69

## Appendix D

### HPC Benchmarks

*Figure D1.* NAS Parallel Benchmark BT

Virtual Nodes - Total	Virtual Nodes - Avg	Physical Nodes - Total	Physical Nodes - Avg	Virtual Nodes - Process	Virtual Nodes - Process Avg	Physical Nodes - Process	Physical Nodes - Process Avg
698.56	694.6117143	630.85	629.9182857	174.64	173.6531	157.71	157.479
701.82	694.6117143	624.77	629.9182857	175.45	173.6531	156.19	157.479
697.93	694.6117143	625.27	629.9182857	174.48	173.6531	156.32	157.479
697.86	694.6117143	626.73	629.9182857	174.47	173.6531	156.68	157.479
700.54	694.6117143	629.95	629.9182857	175.13	173.6531	157.49	157.479
700.5	694.6117143	631.66	629.9182857	175.12	173.6531	157.91	157.479
700.6	694.6117143	629.81	629.9182857	175.15	173.6531	157.45	157.479
699.33	694.6117143	624.53	629.9182857	174.83	173.6531	156.13	157.479
700.12	694.6117143	633.9	629.9182857	175.03	173.6531	158.48	157.479
701.49	694.6117143	627.81	629.9182857	175.37	173.6531	156.95	157.479
688.88	694.6117143	626.22	629.9182857	172.22	173.6531	156.55	157.479
689.17	694.6117143	630.43	629.9182857	172.29	173.6531	157.61	157.479
687.71	694.6117143	631.45	629.9182857	171.93	173.6531	157.86	157.479
688.54	694.6117143	626.52	629.9182857	172.14	173.6531	156.63	157.479
689.1	694.6117143	629.84	629.9182857	172.28	173.6531	157.46	157.479
691.47	694.6117143	629	629.9182857	172.87	173.6531	157.25	157.479
689.99	694.6117143	630.31	629.9182857	172.5	173.6531	157.58	157.479
689.09	694.6117143	627.94	629.9182857	172.27	173.6531	156.99	157.479
694.33	694.6117143	633.85	629.9182857	173.58	173.6531	158.46	157.479
688.02	694.6117143	630.59	629.9182857	172.01	173.6531	157.65	157.479
690.82	694.6117143	630.71	629.9182857	172.71	173.6531	157.68	157.479
691.85	694.6117143	628.58	629.9182857	172.96	173.6531	157.15	157.479
693.08	694.6117143	631.56	629.9182857	173.27	173.6531	157.89	157.479
692.55	694.6117143	631.24	629.9182857	173.14	173.6531	157.81	157.479
689.45	694.6117143	632.67	629.9182857	172.36	173.6531	158.17	157.479
694.54	694.6117143	628.89	629.9182857	173.64	173.6531	157.22	157.479
690.96	694.6117143	627.32	629.9182857	172.74	173.6531	156.83	157.479
694.29	694.6117143	631.84	629.9182857	173.57	173.6531	157.96	157.479
694.38	694.6117143	633.07	629.9182857	173.6	173.6531	158.27	157.479
693.04	694.6117143	630.38	629.9182857	173.26	173.6531	157.59	157.479
698.45	694.6117143	632.05	629.9182857	174.61	173.6531	158.01	157.479
696.78	694.6117143	630.92	629.9182857	174.2	173.6531	157.73	157.479
697.51	694.6117143	633.68	629.9182857	174.38	173.6531	158.42	157.479
700.09	694.6117143	631.3	629.9182857	175.02	173.6531	157.82	157.479
698.57	694.6117143	631.5	629.9182857	174.64	173.6531	157.88	157.479

Figure D2. NAS Parallel Benchmarks CG

Virtual Nodes - Total	Virtual Nodes - Avg	Physical Nodes - Total	Physical Nodes - Avg	Virtual Nodes - Process	Virtual Nodes - Process Avg	Physical Nodes - Process	Physical Nodes - Process Avg
67.49	67.70176471	92.61	91.76857143	16.87	16.925	23.15	22.9417
67.43	67.70176471	92.2	91.76857143	16.86	16.925	23.05	22.9417
67.45	67.70176471	92.36	91.76857143	16.86	16.925	23.09	22.9417
67.66	67.70176471	92.53	91.76857143	16.91	16.925	23.13	22.9417
67.69	67.70176471	91.16	91.76857143	16.92	16.925	22.79	22.9417
67.68	67.70176471	92.08	91.76857143	16.92	16.925	23.02	22.9417
67.72	67.70176471	92.68	91.76857143	16.93	16.925	23.17	22.9417
67.81	67.70176471	92.36	91.76857143	16.95	16.925	23.09	22.9417
67.91	67.70176471	92.71	91.76857143	16.98	16.925	23.18	22.9417
67.94	67.70176471	91.76	91.76857143	16.99	16.925	22.94	22.9417
67.68	67.70176471	90.05	91.76857143	16.92	16.925	22.51	22.9417
67.65	67.70176471	89.66	91.76857143	16.91	16.925	22.42	22.9417
67.73	67.70176471	91.96	91.76857143	16.93	16.925	22.99	22.9417
67.7	67.70176471	91.58	91.76857143	16.93	16.925	22.89	22.9417
67.68	67.70176471	92.07	91.76857143	16.92	16.925	23.02	22.9417
67.73	67.70176471	92.14	91.76857143	16.93	16.925	23.03	22.9417
67.66	67.70176471	92.02	91.76857143	16.91	16.925	23	22.9417
67.72	67.70176471	92.07	91.76857143	16.93	16.925	23.02	22.9417
67.77	67.70176471	89.54	91.76857143	16.94	16.925	22.39	22.9417
67.75	67.70176471	91.62	91.76857143	16.94	16.925	22.9	22.9417
67.83	67.70176471	91.23	91.76857143	16.96	16.925	22.81	22.9417
67.82	67.70176471	92.69	91.76857143	16.96	16.925	23.17	22.9417
67.7	67.70176471	92.31	91.76857143	16.92	16.925	23.08	22.9417
67.76	67.70176471	92.38	91.76857143	16.94	16.925	23.09	22.9417
67.8	67.70176471	92.44	91.76857143	16.95	16.925	23.11	22.9417
67.64	67.70176471	91.25	91.76857143	16.91	16.925	22.81	22.9417
67.76	67.70176471	89.67	91.76857143	16.94	16.925	22.42	22.9417
67.23	67.70176471	92.59	91.76857143	16.81	16.925	23.15	22.9417
67.74	67.70176471	91.87	91.76857143	16.93	16.925	22.97	22.9417
67.73	67.70176471	92.15	91.76857143	16.93	16.925	23.04	22.9417
67.88	67.70176471	89.39	91.76857143	16.97	16.925	22.35	22.9417
67.74	67.70176471	92.02	91.76857143	16.94	16.925	23	22.9417
67.66	67.70176471	92.16	91.76857143	16.91	16.925	23.04	22.9417
67.72	67.70176471	92.13	91.76857143	16.93	16.925	23.03	22.9417



Figure D3. NAS Parallel Benchmarks EP

Virtual Nodes - Total	Virtual Nodes - Avg	Physical Nodes - Total	Physical Nodes - Avg	Virtual Nodes - Process	Virtual Nodes - Process Avg	Physical Nodes - Process	Physical Nodes - Process Avg
41.66	42.085	25.34	25.32914286	10.42	10.52058824	6.34	6.33343
42.23	42.085	25.36	25.32914286	10.56	10.52058824	6.34	6.33343
42.13	42.085	25.36	25.32914286	10.53	10.52058824	6.34	6.33343
42.13	42.085	25.4	25.32914286	10.53	10.52058824	6.35	6.33343
42.06	42.085	25.44	25.32914286	10.52	10.52058824	6.36	6.33343
42.21	42.085	25.35	25.32914286	10.55	10.52058824	6.34	6.33343
42.08	42.085	25.35	25.32914286	10.52	10.52058824	6.34	6.33343
42.18	42.085	25.36	25.32914286	10.54	10.52058824	6.34	6.33343
42.15	42.085	25.36	25.32914286	10.54	10.52058824	6.34	6.33343
42.09	42.085	25.42	25.32914286	10.52	10.52058824	6.36	6.33343
42	42.085	25.44	25.32914286	10.5	10.52058824	6.36	6.33343
42.13	42.085	25.36	25.32914286	10.53	10.52058824	6.34	6.33343
42.12	42.085	25.4	25.32914286	10.53	10.52058824	6.35	6.33343
42.03	42.085	25.35	25.32914286	10.51	10.52058824	6.34	6.33343
41.9	42.085	25.34	25.32914286	10.48	10.52058824	6.33	6.33343
42.24	42.085	25.44	25.32914286	10.56	10.52058824	6.36	6.33343
41.98	42.085	25.44	25.32914286	10.49	10.52058824	6.36	6.33343
42.08	42.085	25.4	25.32914286	10.52	10.52058824	6.35	6.33343
42.06	42.085	25.35	25.32914286	10.51	10.52058824	6.34	6.33343
42.14	42.085	23.74	25.32914286	10.53	10.52058824	5.94	6.33343
42.06	42.085	25.35	25.32914286	10.51	10.52058824	6.34	6.33343
41.9	42.085	25.42	25.32914286	10.48	10.52058824	6.36	6.33343
42.15	42.085	25.34	25.32914286	10.54	10.52058824	6.34	6.33343
42.01	42.085	25.34	25.32914286	10.5	10.52058824	6.33	6.33343
42.29	42.085	25.36	25.32914286	10.57	10.52058824	6.34	6.33343
42.02	42.085	25.36	25.32914286	10.5	10.52058824	6.34	6.33343
42.05	42.085	25.35	25.32914286	10.51	10.52058824	6.34	6.33343
41.98	42.085	25.35	25.32914286	10.5	10.52058824	6.34	6.33343
42.12	42.085	25.42	25.32914286	10.53	10.52058824	6.35	6.33343
42.33	42.085	25.35	25.32914286	10.58	10.52058824	6.34	6.33343
42.03	42.085	25.36	25.32914286	10.51	10.52058824	6.34	6.33343
42.24	42.085	25.44	25.32914286	10.56	10.52058824	6.36	6.33343
41.98	42.085	25.34	25.32914286	10.49	10.52058824	6.34	6.33343
42.13	42.085	25.35	25.32914286	10.53	10.52058824	6.34	6.33343
42.07	42.085	25.39	25.32914286	10.52	10.52058824	6.35	6.33343

Figure D4. NAS Parallel Benchmarks FT

Virtual Nodes - Total	Virtual Nodes - Avg	Physical Nodes - Total	Physical Nodes - Avg	Virtual Nodes - Process	Virtual Nodes - Process Avg	Physical Nodes - Process	Physical Nodes - Process Avg
55.79	55.10558824	117.98	116.3662857	13.95	13.77647	29.49	29.0911
55.98	55.10558824	116.14	116.3662857	14	13.77647	29.04	29.0911
56.28	55.10558824	112.3	116.3662857	14.07	13.77647	28.08	29.0911
56.12	55.10558824	115.63	116.3662857	14.03	13.77647	28.91	29.0911
56.09	55.10558824	99.04	116.3662857	14.02	13.77647	24.76	29.0911
54.3	55.10558824	115.87	116.3662857	13.57	13.77647	28.97	29.0911
54.36	55.10558824	114.01	116.3662857	13.59	13.77647	28.5	29.0911
54.45	55.10558824	103.51	116.3662857	13.61	13.77647	25.88	29.0911
55.42	55.10558824	110.11	116.3662857	13.86	13.77647	27.53	29.0911
54.37	55.10558824	113.95	116.3662857	13.59	13.77647	28.49	29.0911
55.34	55.10558824	122.38	116.3662857	13.84	13.77647	30.59	29.0911
54.43	55.10558824	119.5	116.3662857	13.61	13.77647	29.87	29.0911
54.3	55.10558824	105.02	116.3662857	13.58	13.77647	26.25	29.0911
54.28	55.10558824	115.22	116.3662857	13.57	13.77647	28.8	29.0911
54.25	55.10558824	115.12	116.3662857	13.56	13.77647	28.78	29.0911
54.38	55.10558824	109.35	116.3662857	13.6	13.77647	27.34	29.0911
55.47	55.10558824	127.56	116.3662857	13.87	13.77647	31.89	29.0911
55.32	55.10558824	128.85	116.3662857	13.83	13.77647	32.21	29.0911
55.25	55.10558824	123	116.3662857	13.81	13.77647	30.75	29.0911
55.2	55.10558824	122.85	116.3662857	13.8	13.77647	30.71	29.0911
55.22	55.10558824	109.26	116.3662857	13.8	13.77647	27.31	29.0911
55.11	55.10558824	115.27	116.3662857	13.78	13.77647	28.82	29.0911
55.19	55.10558824	125.71	116.3662857	13.8	13.77647	31.43	29.0911
56.2	55.10558824	114.19	116.3662857	14.05	13.77647	28.55	29.0911
55.84	55.10558824	116.85	116.3662857	13.96	13.77647	29.21	29.0911
56.15	55.10558824	118.6	116.3662857	14.04	13.77647	29.65	29.0911
55.94	55.10558824	116.96	116.3662857	13.99	13.77647	29.24	29.0911
55.81	55.10558824	125	116.3662857	13.95	13.77647	31.25	29.0911
56.01	55.10558824	118.39	116.3662857	14	13.77647	29.6	29.0911
56.06	55.10558824	117.2	116.3662857	14.01	13.77647	29.3	29.0911
56.17	55.10558824	117.46	116.3662857	14.04	13.77647	29.36	29.0911
56.36	55.10558824	108.52	116.3662857	14.09	13.77647	27.13	29.0911
54.66	55.10558824	120.49	116.3662857	13.66	13.77647	30.12	29.0911
54.57	55.10558824	125.26	116.3662857	13.64	13.77647	31.31	29.0911
54.66	55.10558824	116.27	116.3662857	13.66	13.77647	29.07	29.0911

Figure D5. NAS Parallel Benchmarks IS

Virtual Nodes - Total	Virtual Nodes - Avg	Physical Nodes - Total	Physical Nodes - Avg	Virtual Nodes - Process	Virtual Nodes - Process Avg	Physical Nodes - Process	Physical Nodes - Process Avg
1.5	1.554375	4.23	4.274285714	0.38	0.388125	1.06	1.068
1.53	1.554375	4.35	4.274285714	0.38	0.388125	1.09	1.068
1.56	1.554375	4.24	4.274285714	0.39	0.388125	1.06	1.068
1.57	1.554375	4.32	4.274285714	0.39	0.388125	1.08	1.068
1.53	1.554375	4.26	4.274285714	0.38	0.388125	1.07	1.068
1.54	1.554375	4.35	4.274285714	0.38	0.388125	1.09	1.068
1.53	1.554375	4.26	4.274285714	0.38	0.388125	1.07	1.068
1.53	1.554375	4.22	4.274285714	0.38	0.388125	1.05	1.068
1.5	1.554375	4.22	4.274285714	0.38	0.388125	1.05	1.068
1.55	1.554375	4.3	4.274285714	0.39	0.388125	1.07	1.068
1.56	1.554375	4.32	4.274285714	0.39	0.388125	1.08	1.068
1.55	1.554375	4.17	4.274285714	0.39	0.388125	1.04	1.068
1.56	1.554375	4.23	4.274285714	0.39	0.388125	1.06	1.068
1.54	1.554375	4.35	4.274285714	0.39	0.388125	1.09	1.068
1.53	1.554375	4.26	4.274285714	0.38	0.388125	1.06	1.068
1.57	1.554375	4.33	4.274285714	0.39	0.388125	1.08	1.068
1.58	1.554375	4.37	4.274285714	0.39	0.388125	1.09	1.068
1.56	1.554375	4.27	4.274285714	0.39	0.388125	1.07	1.068
1.6	1.554375	4.24	4.274285714	0.4	0.388125	1.06	1.068
1.56	1.554375	4.26	4.274285714	0.39	0.388125	1.06	1.068
1.57	1.554375	4.27	4.274285714	0.39	0.388125	1.07	1.068
1.6	1.554375	4.26	4.274285714	0.4	0.388125	1.07	1.068
1.56	1.554375	4.27	4.274285714	0.39	0.388125	1.07	1.068
1.56	1.554375	4.26	4.274285714	0.39	0.388125	1.06	1.068
1.59	1.554375	4.25	4.274285714	0.4	0.388125	1.06	1.068
1.6	1.554375	4.25	4.274285714	0.4	0.388125	1.06	1.068
1.58	1.554375	4.35	4.274285714	0.39	0.388125	1.09	1.068
1.53	1.554375	4.25	4.274285714	0.38	0.388125	1.06	1.068
1.57	1.554375	4.26	4.274285714	0.39	0.388125	1.06	1.068
1.56	1.554375	4.26	4.274285714	0.39	0.388125	1.06	1.068
1.53	1.554375	4.25	4.274285714	0.38	0.388125	1.06	1.068
1.54	1.554375	4.32	4.274285714	0.39	0.388125	1.08	1.068

Figure D6. NAS Parallel Benchmarks LU

Virtual Nodes - Total	Virtual Nodes - Avg	Physical Nodes - Total	Physical Nodes - Avg	Virtual Nodes - Process	Virtual Nodes - Process Avg	Physical Nodes - Process	Physical Nodes - Process Avg
874.66	873.986	831.76	832.7982857	218.67	218.4963	207.94	208.199
874.82	873.986	830	832.7982857	218.71	218.4963	207.5	208.199
875.48	873.986	831.14	832.7982857	218.87	218.4963	207.78	208.199
874.34	873.986	828.14	832.7982857	218.58	218.4963	207.03	208.199
870.11	873.986	832.39	832.7982857	217.53	218.4963	208.1	208.199
875.02	873.986	831.08	832.7982857	218.76	218.4963	207.77	208.199
873.7	873.986	829.79	832.7982857	218.43	218.4963	207.45	208.199
874.59	873.986	834.92	832.7982857	218.65	218.4963	208.73	208.199
869.95	873.986	833.44	832.7982857	217.49	218.4963	208.36	208.199
874.36	873.986	831.49	832.7982857	218.59	218.4963	207.87	208.199
870.02	873.986	837.55	832.7982857	217.51	218.4963	209.39	208.199
871.45	873.986	833.59	832.7982857	217.86	218.4963	208.4	208.199
869.14	873.986	836.5	832.7982857	217.28	218.4963	209.13	208.199
870.91	873.986	833.86	832.7982857	217.73	218.4963	208.46	208.199
844.06	873.986	835.38	832.7982857	211.01	218.4963	208.85	208.199
870	873.986	835.74	832.7982857	217.5	218.4963	208.93	208.199
872.51	873.986	825.16	832.7982857	218.13	218.4963	206.29	208.199
870.08	873.986	835.44	832.7982857	217.52	218.4963	208.86	208.199
845.64	873.986	835.62	832.7982857	211.41	218.4963	208.9	208.199
859.44	873.986	835.54	832.7982857	214.86	218.4963	208.89	208.199
868.62	873.986	831.13	832.7982857	217.16	218.4963	207.78	208.199
866.1	873.986	832.8	832.7982857	216.53	218.4963	208.2	208.199
870.45	873.986	830.46	832.7982857	217.61	218.4963	207.61	208.199
844.47	873.986	832.06	832.7982857	211.12	218.4963	208.01	208.199
861.68	873.986	831.18	832.7982857	215.42	218.4963	207.8	208.199
865.13	873.986	831.87	832.7982857	216.28	218.4963	207.97	208.199
865.18	873.986	833.76	832.7982857	216.29	218.4963	208.44	208.199
861.77	873.986	829.58	832.7982857	215.44	218.4963	207.39	208.199
880.97	873.986	835.86	832.7982857	220.24	218.4963	208.97	208.199
872.96	873.986	832.97	832.7982857	218.24	218.4963	208.24	208.199
871.58	873.986	836.53	832.7982857	217.9	218.4963	209.13	208.199
874.41	873.986	831.87	832.7982857	218.6	218.4963	207.97	208.199
867.29	873.986	832.77	832.7982857	216.82	218.4963	208.19	208.199
872.42	873.986	834.91	832.7982857	218.1	218.4963	208.73	208.199
876.46	873.986	831.66	832.7982857	219.11	218.4963	207.91	208.199

Figure D7. NAS Parallel Benchmarks MG

Virtual Nodes - Total	Virtual Nodes - Avg	Physical Nodes - Total	Physical Nodes - Avg	Virtual Nodes - Process	Virtual Nodes - Process Avg	Physical Nodes - Process	Physical Nodes - Process Avg
233.53	233.6967647	467.19	467.2228571	58.38	58.4247	116.8	116.806
232.95	233.6967647	465.47	467.2228571	58.24	58.4247	116.37	116.806
232.98	233.6967647	467.97	467.2228571	58.25	58.4247	116.99	116.806
232.42	233.6967647	465.89	467.2228571	58.11	58.4247	116.47	116.806
233.81	233.6967647	466.36	467.2228571	58.45	58.4247	116.59	116.806
232.26	233.6967647	464.05	467.2228571	58.07	58.4247	116.01	116.806
233.72	233.6967647	470.19	467.2228571	58.43	58.4247	117.55	116.806
234.86	233.6967647	466.08	467.2228571	58.72	58.4247	116.52	116.806
234.17	233.6967647	467.81	467.2228571	58.54	58.4247	116.95	116.806
233.67	233.6967647	468.02	467.2228571	58.42	58.4247	117.01	116.806
234.4	233.6967647	467.76	467.2228571	58.6	58.4247	116.94	116.806
234.96	233.6967647	465.56	467.2228571	58.74	58.4247	116.39	116.806
231.95	233.6967647	468.54	467.2228571	57.99	58.4247	117.14	116.806
234.85	233.6967647	468.24	467.2228571	58.71	58.4247	117.06	116.806
232.56	233.6967647	466.75	467.2228571	58.14	58.4247	116.69	116.806
233.01	233.6967647	467	467.2228571	58.25	58.4247	116.75	116.806
234.11	233.6967647	468.84	467.2228571	58.53	58.4247	117.21	116.806
234.18	233.6967647	466.57	467.2228571	58.54	58.4247	116.64	116.806
232.12	233.6967647	468.38	467.2228571	58.03	58.4247	117.09	116.806
233.07	233.6967647	464.33	467.2228571	58.27	58.4247	116.08	116.806
234.07	233.6967647	470.27	467.2228571	58.52	58.4247	117.57	116.806
234.29	233.6967647	467.55	467.2228571	58.57	58.4247	116.89	116.806
233.39	233.6967647	468.01	467.2228571	58.35	58.4247	117	116.806
233.81	233.6967647	467.97	467.2228571	58.45	58.4247	116.99	116.806
233.8	233.6967647	467.71	467.2228571	58.45	58.4247	116.93	116.806
232.93	233.6967647	468.92	467.2228571	58.23	58.4247	117.23	116.806
233.49	233.6967647	467.08	467.2228571	58.37	58.4247	116.77	116.806
233.62	233.6967647	465.92	467.2228571	58.4	58.4247	116.48	116.806
233.47	233.6967647	465.72	467.2228571	58.37	58.4247	116.43	116.806
234.24	233.6967647	468.34	467.2228571	58.56	58.4247	117.09	116.806
234.26	233.6967647	466.78	467.2228571	58.57	58.4247	116.7	116.806
235.69	233.6967647	467.33	467.2228571	58.92	58.4247	116.83	116.806
233.55	233.6967647	464.47	467.2228571	58.39	58.4247	116.12	116.806
234.16	233.6967647	467.35	467.2228571	58.54	58.4247	116.84	116.806
234.87	233.6967647	468.38	467.2228571	58.72	58.4247	117.09	116.806

Figure D8. NAS Parallel Benchmarks SP

Virtual Nodes - Total	Virtual Nodes - Avg	Physical Nodes - Total	Physical Nodes - Avg	Virtual Nodes - Process	Virtual Nodes - Process Avg	Physical Nodes - Process	Physical Nodes - Process Avg
272.43	270.1773529	334.15	334.8657143	68.11	67.54470588	83.54	83.72617647
271.43	270.1773529	335.36	334.8657143	67.86	67.54470588	83.84	83.72617647
273.52	270.1773529	338.31	334.8657143	68.38	67.54470588	84.58	83.72617647
258.23	270.1773529	335.38	334.8657143	64.56	67.54470588	83.84	83.72617647
270.88	270.1773529	336.37	334.8657143	67.72	67.54470588	84.09	83.72617647
273.2	270.1773529	334.5	334.8657143	68.3	67.54470588	83.62	83.72617647
271.97	270.1773529	329.38	334.8657143	67.99	67.54470588	82.35	83.72617647
273.35	270.1773529	334.39	334.8657143	68.34	67.54470588	83.6	83.72617647
272.44	270.1773529	334.36	334.8657143	68.11	67.54470588	83.59	83.72617647
272.43	270.1773529	335.45	334.8657143	68.11	67.54470588	83.86	83.72617647
273.02	270.1773529	333.81	334.8657143	68.25	67.54470588	83.45	83.72617647
270.7	270.1773529	334.59	334.8657143	67.68	67.54470588	83.65	83.72617647
270.56	270.1773529	336.36	334.8657143	67.64	67.54470588	84.09	83.72617647
272.18	270.1773529	334.89	334.8657143	68.05	67.54470588	83.72	83.72617647
269.9	270.1773529	335.24	334.8657143	67.48	67.54470588	83.81	83.72617647
271.76	270.1773529	334.89	334.8657143	67.94	67.54470588	83.72	83.72617647
270.91	270.1773529	334.65	334.8657143	67.73	67.54470588	83.66	83.72617647
270.31	270.1773529	335.5	334.8657143	67.58	67.54470588	83.88	83.72617647
270.81	270.1773529	331.13	334.8657143	67.7	67.54470588	82.78	83.72617647
271.95	270.1773529	336.46	334.8657143	67.99	67.54470588	84.11	83.72617647
269.19	270.1773529	334.52	334.8657143	67.3	67.54470588	83.63	83.72617647
270.25	270.1773529	335.86	334.8657143	67.56	67.54470588	83.97	83.72617647
271.18	270.1773529	334.62	334.8657143	67.79	67.54470588	83.66	83.72617647
271.26	270.1773529	333.63	334.8657143	67.82	67.54470588	83.41	83.72617647
255.82	270.1773529	335.16	334.8657143	63.96	67.54470588	83.79	83.72617647
271.33	270.1773529	334.69	334.8657143	67.83	67.54470588	83.67	83.72617647
269.35	270.1773529	335.54	334.8657143	67.34	67.54470588	83.89	83.72617647
270.84	270.1773529	334.42	334.8657143	67.71	67.54470588	83.6	83.72617647
270.05	270.1773529	335.81	334.8657143	67.51	67.54470588	83.95	83.72617647
271	270.1773529	337.35	334.8657143	67.75	67.54470588	84.34	83.72617647
270.65	270.1773529	334.55	334.8657143	67.66	67.54470588	83.64	83.72617647
270.86	270.1773529	335.2	334.8657143	67.71	67.54470588	83.8	83.72617647
265.09	270.1773529	335.79	334.8657143	66.27	67.54470588	83.95	83.72617647
267.18	270.1773529	334.45	334.8657143	66.79	67.54470588	83.61	83.72617647

Figure D9. XHPL

Virtual Nodes	Virtual Nodes Avg	Physical Nodes	Physical Nodes Avg
8.35E-001	8.41E-001	1.80E+000	1.80E+000
8.39E-001	8.41E-001	1.80E+000	1.80E+000
8.38E-001	8.41E-001	1.80E+000	1.80E+000
8.39E-001	8.41E-001	1.80E+000	1.80E+000
8.37E-001	8.41E-001	1.80E+000	1.80E+000
8.39E-001	8.41E-001	1.80E+000	1.80E+000
8.38E-001	8.41E-001	1.80E+000	1.80E+000
8.39E-001	8.41E-001	1.80E+000	1.80E+000
8.37E-001	8.41E-001	1.80E+000	1.80E+000
8.40E-001	8.41E-001	1.80E+000	1.80E+000
8.42E-001	8.41E-001	1.80E+000	1.80E+000
8.39E-001	8.41E-001	1.80E+000	1.80E+000
8.41E-001	8.41E-001	1.80E+000	1.80E+000
8.42E-001	8.41E-001	1.80E+000	1.80E+000
8.40E-001	8.41E-001	1.80E+000	1.80E+000
8.38E-001	8.41E-001	1.80E+000	1.80E+000
8.40E-001	8.41E-001	1.78E+000	1.80E+000
8.41E-001	8.41E-001	1.80E+000	1.80E+000
8.39E-001	8.41E-001	1.80E+000	1.80E+000
8.41E-001	8.41E-001	1.80E+000	1.80E+000
8.41E-001	8.41E-001	1.80E+000	1.80E+000
8.42E-001	8.41E-001	1.78E+000	1.80E+000
8.42E-001	8.41E-001	1.80E+000	1.80E+000
8.40E-001	8.41E-001	1.80E+000	1.80E+000
8.45E-001	8.41E-001	1.80E+000	1.80E+000
8.41E-001	8.41E-001	1.80E+000	1.80E+000
8.43E-001	8.41E-001	1.80E+000	1.80E+000
8.45E-001	8.41E-001	1.80E+000	1.80E+000
8.45E-001	8.41E-001	1.80E+000	1.80E+000
8.43E-001	8.41E-001	1.80E+000	1.80E+000
8.43E-001	8.41E-001	1.80E+000	1.80E+000
8.43E-001	8.41E-001	1.80E+000	1.80E+000
8.42E-001	8.41E-001	1.80E+000	1.80E+000
8.46E-001	8.41E-001	1.80E+000	1.80E+000
8.44E-001	8.41E-001	1.80E+000	1.80E+000